Data Security

Lect. 10,11,12

Network security protocols in practice

Attacks on Internet infrastructure:

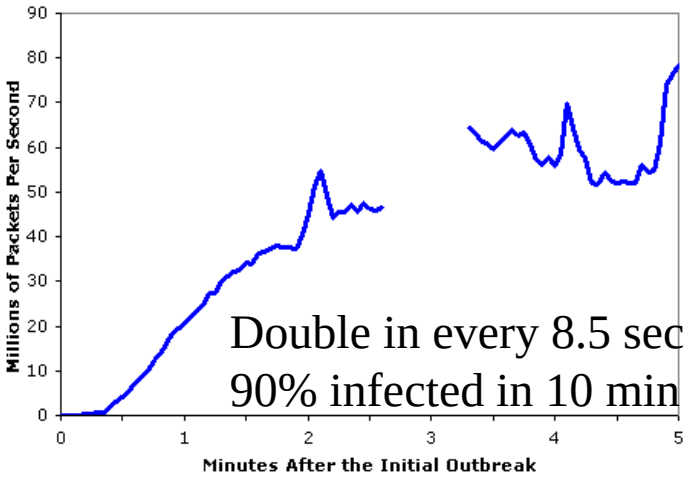
- Infecting/Attacking hosts: spyware, virus, worms, Trojan Horse, unauthorized access, and malware in general.
 - Malware (Malicious software): The word malware is a combination of two words "malicious" and "software". It is a generic term used to describe all of the hostile and intrusive program codes including viruses, spyware, worms, Trojans, or anything that is designed to perform malicious operations on a computer..
 - In law, malware is defined as a computer contaminant
- Denial of service: deny access to resources (servers)
 - server flooding.

Types of Malicious Code (malware)

- Spyware:
 - infection by downloading web page with spyware
 - records web sites visited, upload info to collection site
- Virus:
 - infection by receiving object (e.g., e-mail attachment), actively executing
 - self-replicating: propagate itself to other hosts, users

- Worm:
 - infection by passively receiving object that gets itself executed.
 - self- replicating: propagates to other hosts, users

Sapphire Worm in 2003: aggregate scans/sec in first 5 minutes of outbreak (CAIDA, UWisc data)



Virus - A hidden, *self-replicating section of computer software*, that *propagates by infecting* (i.e., inserting a copy of itself into and *becoming part of) another program*.

A virus cannot run by itself; it requires that its host program be run to make the virus active.

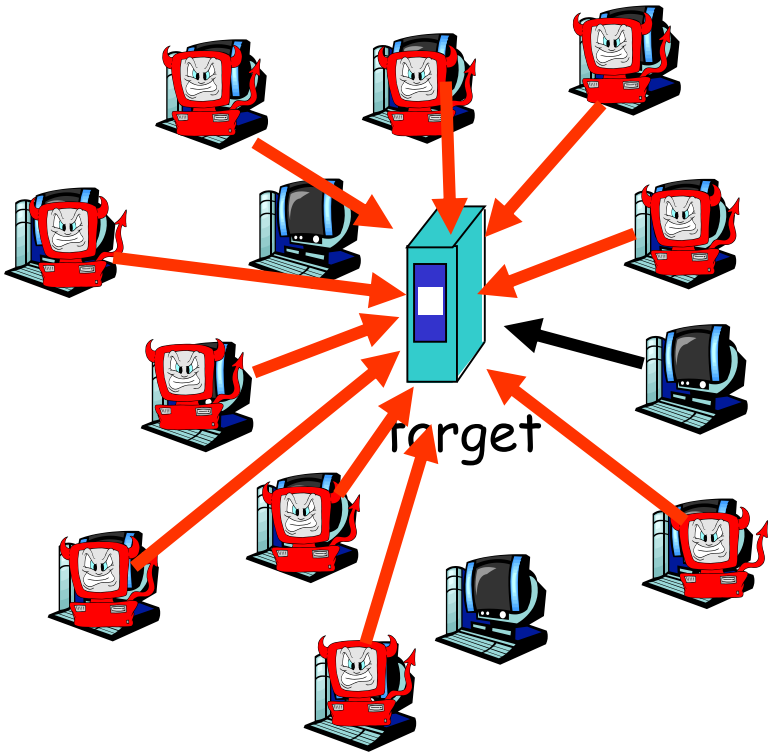
Worm - A computer *program* that can run independently, *can propagate a complete working version of itself* onto other hosts on a network, and may consume computer resources destructively.

.

Trojan horse - A computer *program that appears to have a useful function*, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the program.

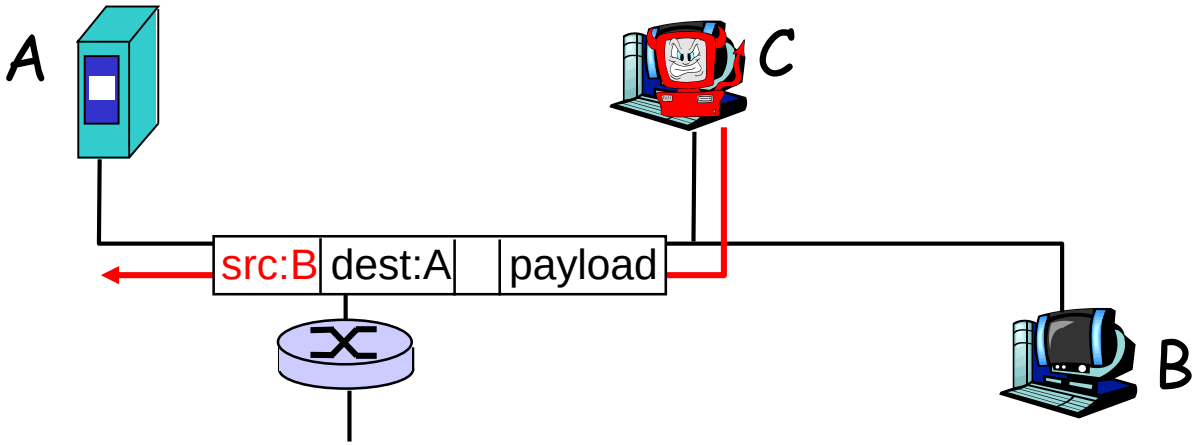
Denial of service attacks:

- Attackers make resources (server) unavailable to legitimate traffic by overwhelming resource with bogus traffic



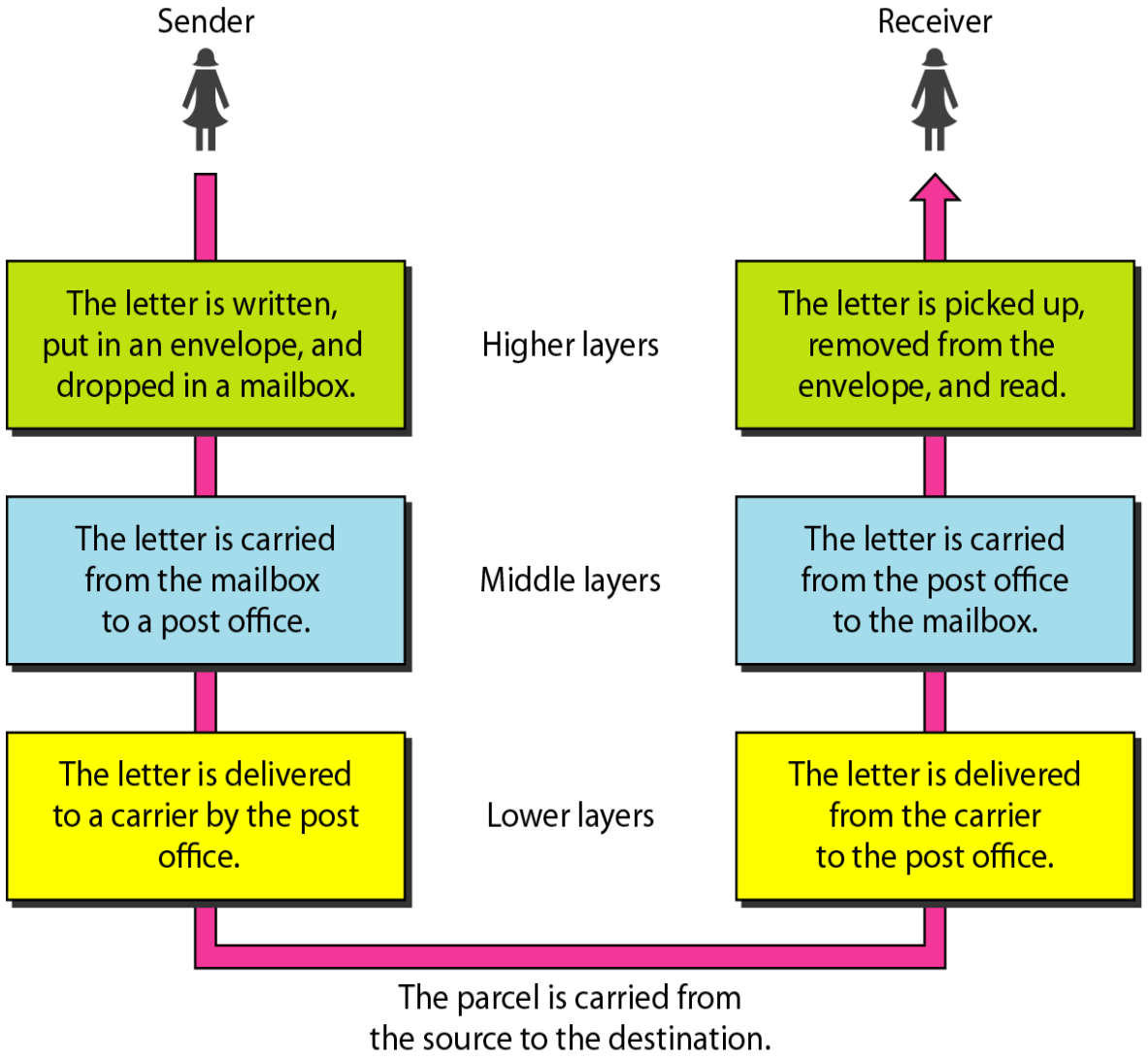
Masquerade as you

- *IP spoofing:* send packet with false source address



Internet (Network) Security Protocols and Standards

Figure Tasks involved in sending a letter



Review of seven layers (OSI-model)

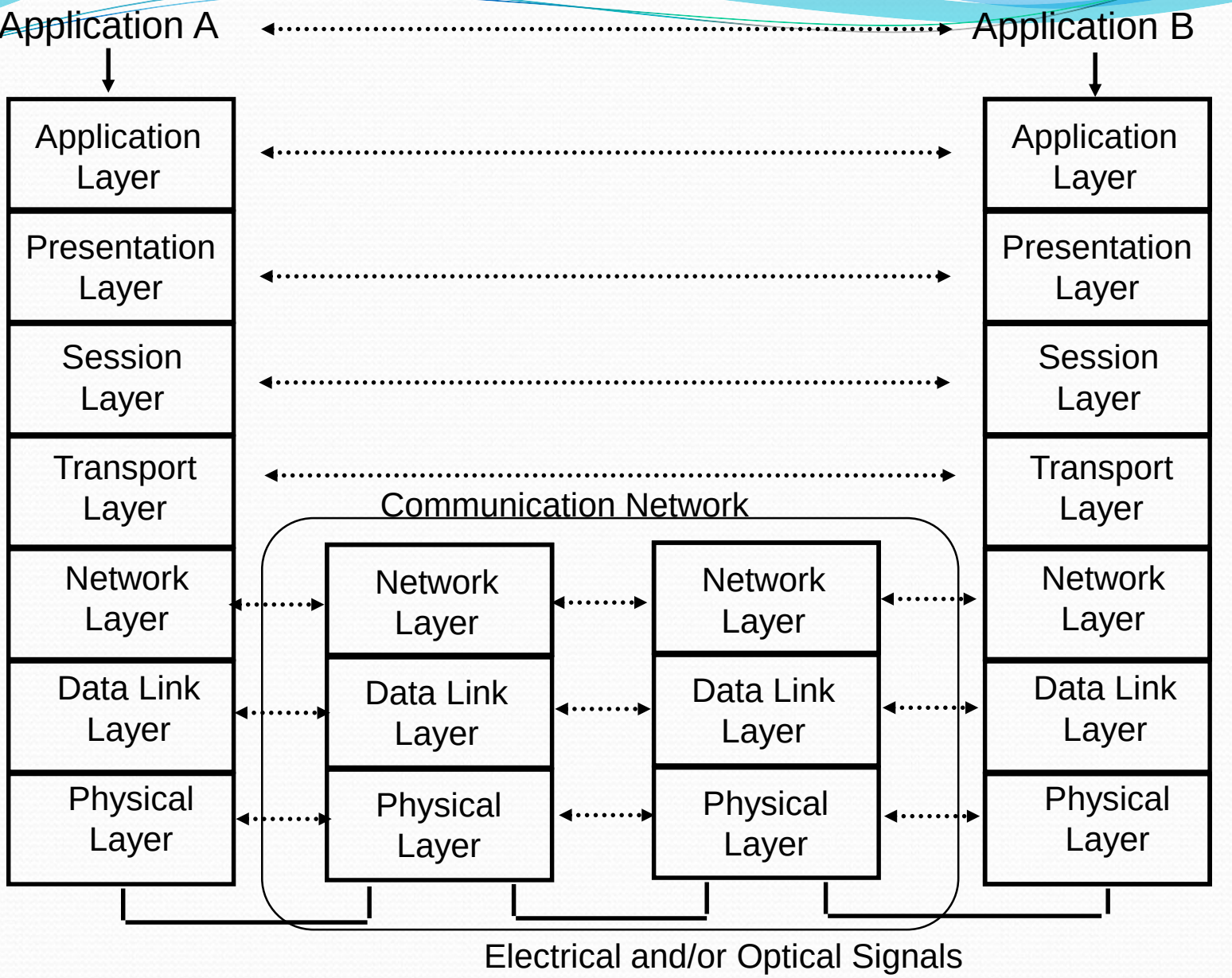


Figure 2.6

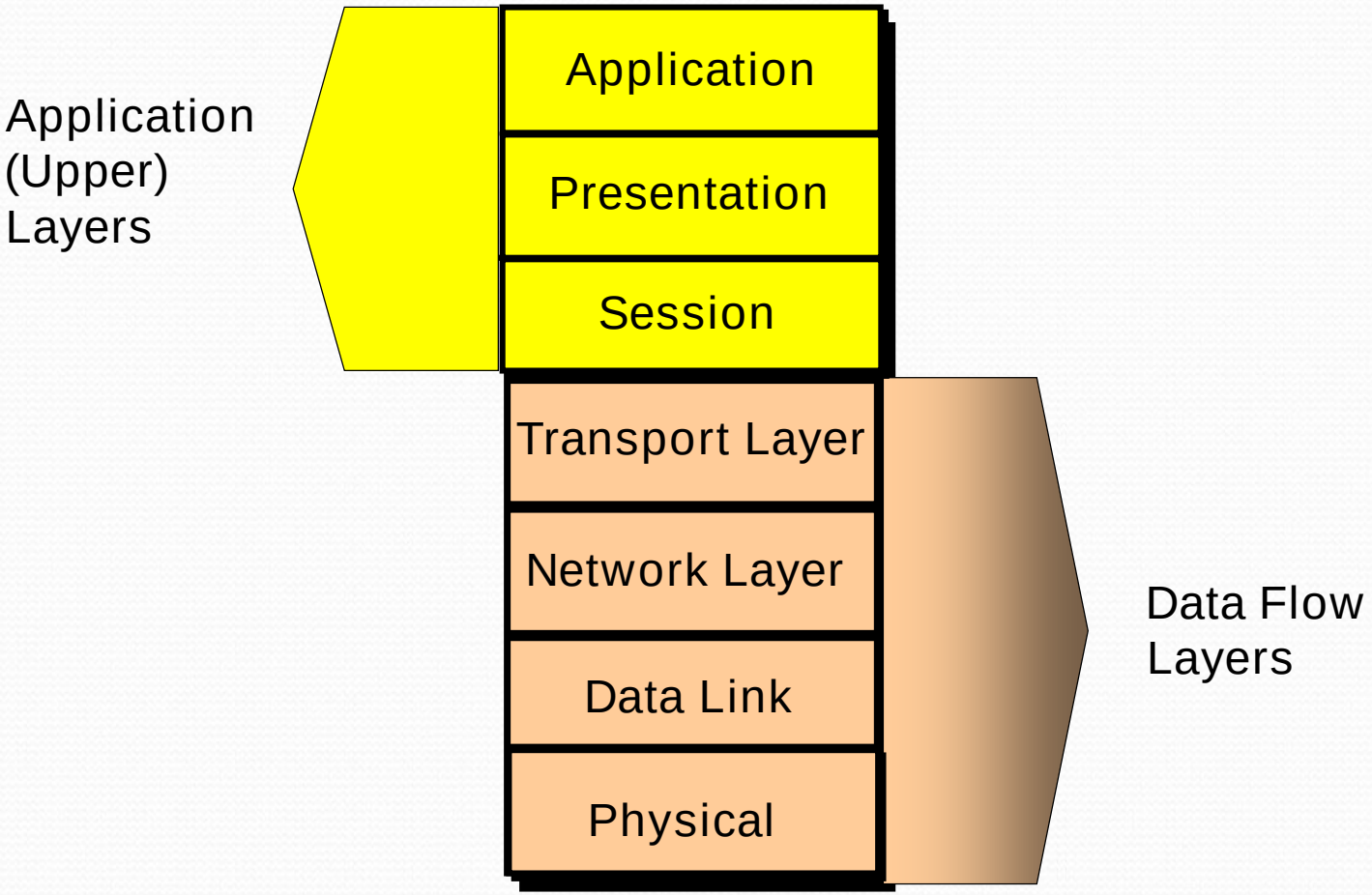
- OSI Open System Interconnection

We want the system to be standard to can any one use it

- ISO International Standard Organization

Make standarization to main task for operation (IEEE)

OSI Model Overview



1. Application Layer

The communicating processes themselves and the actual 'content' transmitted.

- HTTP : Browsing protocol
- FTP : File Transfer Protocol
- Telnet : Remote access protocol
- SMTP : Simple Mail Transfer protocol
- SNMP : Simple Network Management Protocol

2. Presentation layer

- Format of data
- Compression & decompression of data
- Encoding & decoding of data

Such ac ASCII code (8bit) , EBCDIC code (5bit)

3. Session layer

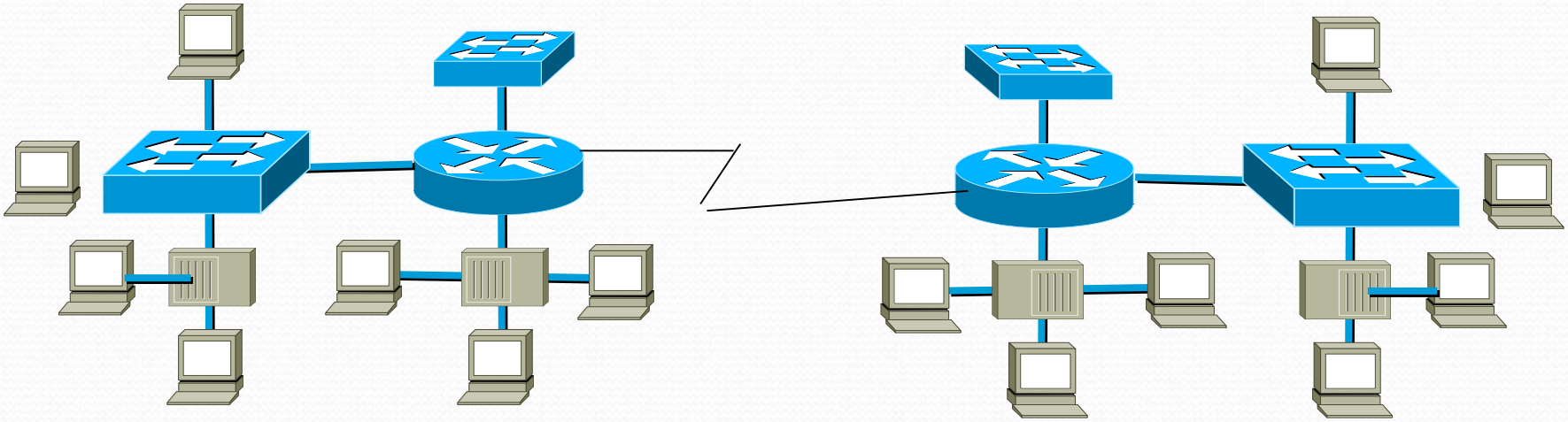
- Set a logical connection (session) between different application
- Specifies communication mode

(simple - Half duplex – full duplex)

4- Transport layer

- Flow control
- Error recovery (reliability)
 (TCP/UDP)

5-Network layer



1-routing

2-IP addressing

2- Data link

- 1- Arbitration : find the best time to send the data (CSMA/CD & taken)

- Error detection

1- parity check : but very week

2-CRC : Depend on mathematical equation

- (Ethernet/WiFi):
 transmission of frames over a single hop

(TCP/IP model-5 layers)

- Application
- Transport
- Network
- Data link
- Physical

TCP (connection-oriented)

The **Transmission Control Protocol** (**TCP**) is one of the main protocols of the Internet protocol suite. It originated in the initial network implementation in which it complemented the Internet Protocol (IP). Therefore, the entire suite is commonly referred to as *TCP/IP*. TCP provides reliable, ordered, and error-checked delivery of a data.

Major Internet applications such as the World Wide Web, email, remote administration, and file transfer rely on TCP. Applications that do not require reliable data stream service may use the User Datagram Protocol (UDP), which provides a connectionless datagram service that emphasizes reduced latency over reliability.

UDP(connectionless)

- User Datagram Protocol uses a simple connectionless transmission model. UDP provides checksums for data integrity, and port numbers for addressing different functions at the source and destination of the datagram. It has no handshaking dialogues. It is unreliable.
- so there is no guarantee of delivery, ordering, or duplicate protection.
- UDP is suitable for purposes where error checking and correction is either not necessary or is performed in the application, avoiding the overhead of such processing at the network interface level. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for delayed packets, which may not be an option in a real-time system.
- The *lack of retransmission delays* makes it suitable for real-time applications such as Voice over IP, online games, and many protocols .

Internet Security Protocols

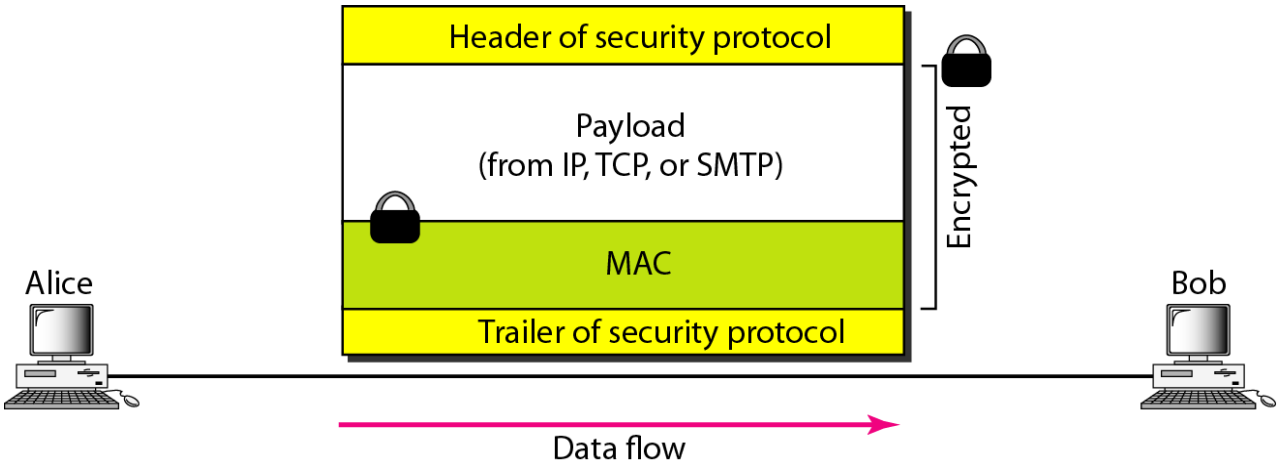
- Application Layer:
 - PGP
 - S/MIME
 - HTTPS
- Transport Layer:
 - SSL
 - TLS
- Network Layer:
 - IPSec
 - VPN
- Data Link Layer:
 - PPP
 - IEEE 802.11 (WEP, WPA)

Generally…

- When security is placed at lower levels, it can provide automatic, “blanket” coverage…
 - …but it can take a long time before it is widely adopted
 - Can be inefficient to encrypt everything
- When security is placed at higher levels, individual users can choose when to use it…
 - …but users who are not security-conscious may not take advantage of it
 - Can encrypt only what is necessary

Security in the Internet: SSL/TLS, PGP, IPSec, VPN

Common structure of three security protocols(SSL , IPSEC, PGP)



In all these protocols, there are some common issues that we need to consider. First, we need to create a MAC. Then we need to encrypt the message and, probably, the MAC. This means, that with some minor variations, the three protocols discussed in this chapter take a packet from the appropriate layer and create a new packet which is authenticated and encrypted.

One common issue in all these protocols is security parameters. The sender and the receiver, before they can send secured data to each other, they need to know which algorithms to use for authentication and encryption/decryption. Bob and Alice still need at least two keys: one for the MAC and one for encryption/decryption.

To send secured data, we need a set of security parameters. To limit the steps need for the exchange of these parameters, we can use public-key cryptography if each person has a private and public key pair. The number of steps can be reduced to one or two.

In the one-step version, we can use session keys to create the MAC and encrypt both data and MAC. The session keys and the list of algorithms can be sent with the packet but encrypted by using public-key ciphers.

In the two-step version, we first establish the security parameters by using public-key ciphers. We then use the security parameters to securely send actual data. One of the three protocols, PGP, uses the first approach; the other two protocols, IPSec and SSL/TLS, use the second.

Internet Services

- Two of the most popular—
 - The World Wide Web and
 - Email.

Broadly speaking, both of these applications use the request/reply paradigm—users send requests to servers, which then respond accordingly.

- It is important to distinguish between application programs and application protocols.
- For example, the HyperText Transport Protocol (HTTP) is an application protocol that is used to retrieve Web pages from remote servers.
- There can be many different application programs—that is, Web clients like Internet Explorer, Chrome, Firefox, and Safari—that provide users with a different look and feel, but all of them use the same HTTP protocol to communicate with Web servers over the Internet.

- Two very widely-used, standardized application protocols:
 - SMTP: Simple Mail Transfer Protocol is used to exchange electronic mail.
 - HTTP: HyperText Transport Protocol is used to communicate between Web browsers and Web servers.

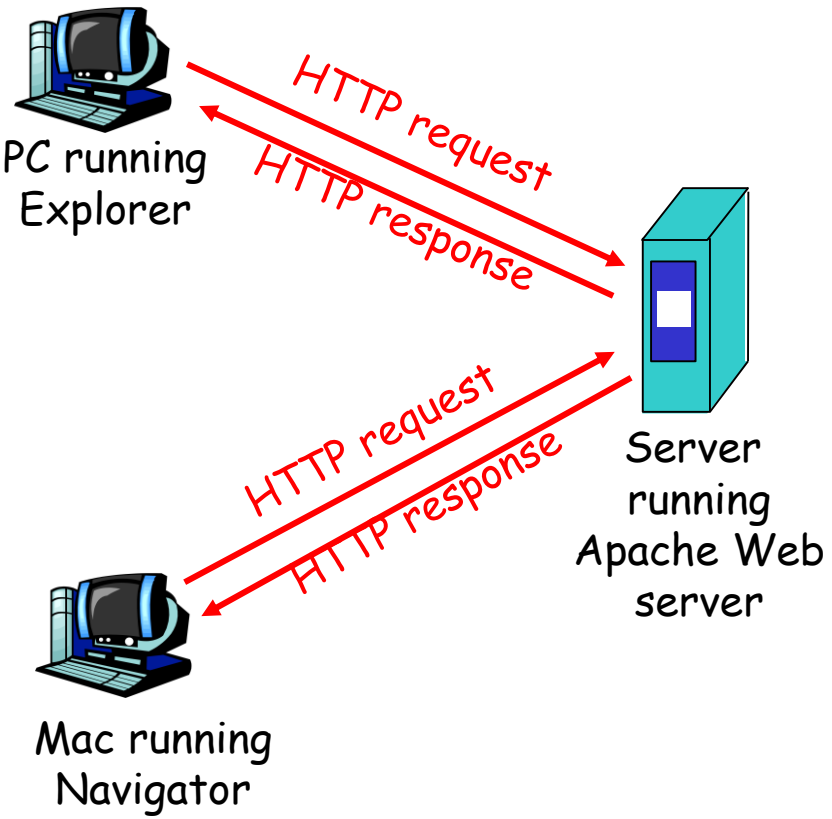
- World Wide Web
 - The core idea of hypertext is that one document can link to another document, and the protocol (HTTP) and document language (HTML) were designed to meet that goal.
 - One helpful way to think of the Web is as a set of cooperating clients and servers, all of whom speak the same language: HTTP.
 - Most people are exposed to the Web through a graphical client program, or Web browser, like Safari, Chrome, Firefox or Internet Explorer.

- World Wide Web
 - Hence, any Web browser has a function that allows the user to obtain an object by "opening a URL."
 - URLs (Uniform Resource Locators) are so familiar to most of us by now that it's easy to forget that they haven't been around forever.
 - They provide information that allows objects on the Web to be located, and they look like the following:
 - http://www.cs.princeton.edu/index.html

- World Wide Web
 - If you opened that particular URL, your Web browser would open a TCP connection to the Web server at a machine called www.cs.princeton.edu and immediately retrieve and display the file called index.html.
 - Most files on the Web contain images and text and many have other objects such as audio and video clips, pieces of code, etc.
 - They also frequently include URLs that point to other files that may be located on other machines, which is the core of the “hypertext” part of HTTP and HTML.

HTTP: hypertext transfer protocol

- client/server model
 - client: browser that requests, receives, "displays" Web objects
 - server: Web server sends objects in response to requests
- HTTP 1.0: RFC 1945
- HTTP 1.1: RFC 2068



- Web page consists of base HTML-file which includes several referenced objects
- Object can be HTML file, JPEG image, Java applet, audio file,…
- Each object is addressable by a URL
- Example URL:



2-Electronic Mail (SMTP, MIME, IMAP)

- Email is one of the oldest network applications
- It is important:
- (1) to distinguish the user interface (i.e., your mail reader) from the underlying message transfer protocols (such as SMTP or IMAP), and
- (2) to distinguish between this transfer protocol and a companion protocol (RFC 822 and MIME) that defines the format of the messages being exchanged.

Electronic Mail (SMTP, MIME, IMAP)

Message Transfer

- For many years, the majority of email was moved from host to host using only SMTP.
- While SMTP continues to play a central role, it is now just one email protocol of several,
- IMAP and POP being two other important protocols for retrieving mail messages.

Electronic Mail (SMTP, MIME, IMAP)

Message Transfer

- First, users interact with a *mail reader when they compose, file, search, and read their email.*
 - There are countless mail readers available, just like there are many Web browsers to choose from.
 - In the early days of the Internet, users typically logged into the machine on which their mailbox resided, and the mail reader they invoked was a local application program that extracted messages from the file system.
 - Today, of course, users remotely access their mailbox from their laptop or smartphone; they do not first log into the host that stores their mail (a mail server).

Electronic Mail (SMTP, MIME, IMAP)

Message Transfer

- Second, there is a *mail daemon (or process) running on each host that holds a mailbox.*
 - You can think of this process, also called a *message transfer agent (MTA), as* playing the role of a post office: users (or their mail readers) give the daemon messages they want to send to other users, the daemon uses SMTP running over TCP to transmit the message to a daemon running on another machine, and the daemon puts incoming messages into the user's mailbox (where that user's mail reader can later find it).
 - Since SMTP is a protocol that anyone could implement, in theory there could be many different implementations of the mail daemon.

Electronic Mail (SMTP, MIME, IMAP)

Message Transfer

- While it is certainly possible that the MTA on a sender's machine establishes an SMTP/TCP connection to the MTA on the recipient's mail server, in many cases the mail traverses one or more *mail gateways on its route from the sender's host to the* receiver's host.
- Like the end hosts, these gateways also run a message transfer agent process.

- Electronic Mail (SMTP, MIME, IMAP)
 - Mail Reader
 - Today, most often the user accesses his or her mailbox from a remote machine using yet another protocol, such as the Post Office Protocol (POP) or the Internet Message Access Protocol (IMAP).

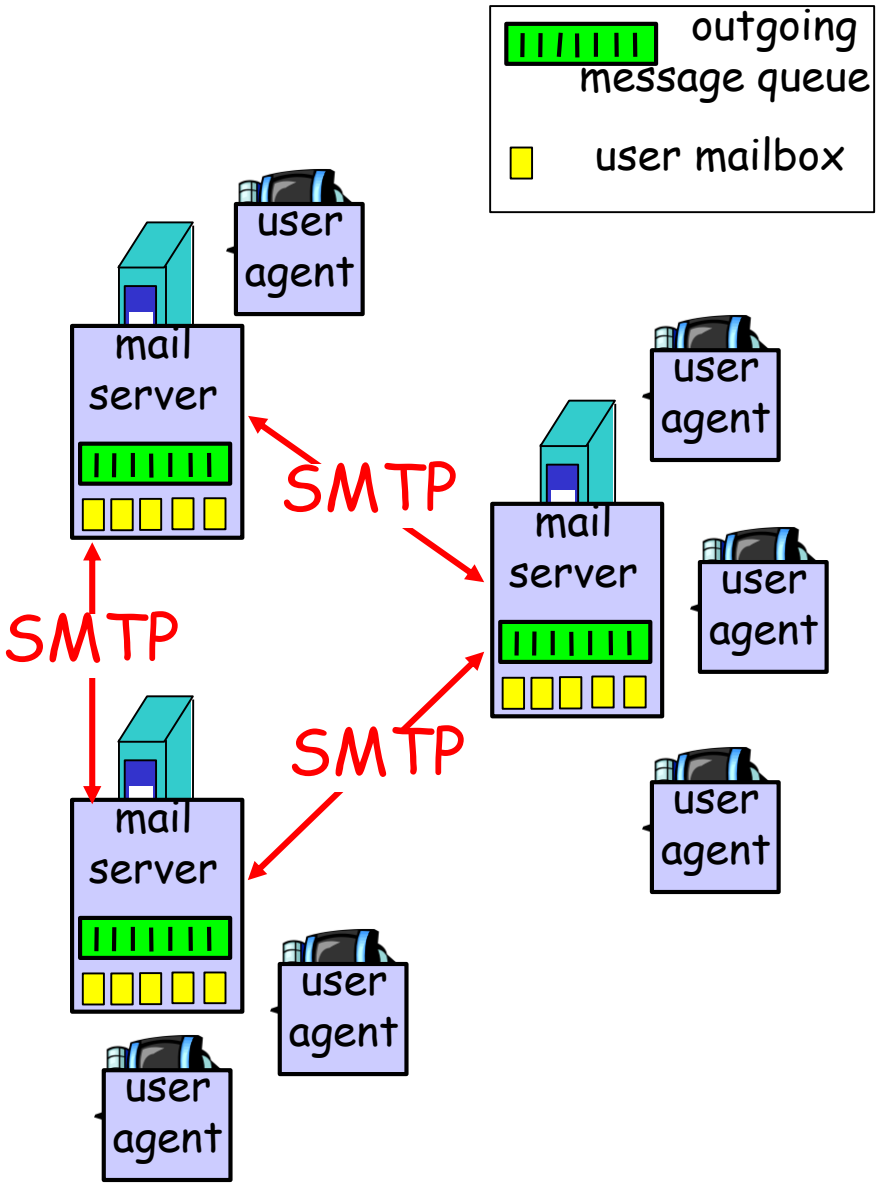
Electronic Mail

Three major components:

- user agents
- mail servers
- simple mail transfer protocol: SMTP

User Agent (Email Reader)

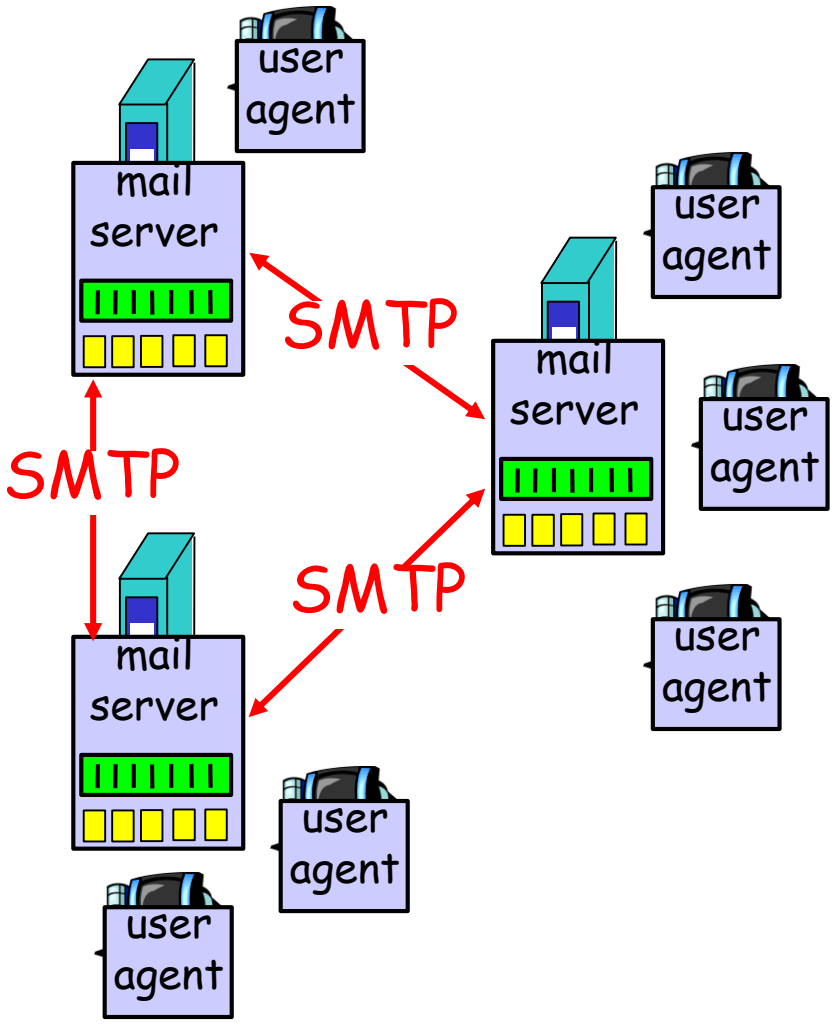
- a.k.a. "mail reader"
- composing, editing, reading mail messages
- e.g., Eudora, Outlook, elm, Netscape Messenger
- outgoing, incoming messages stored on server



Electronic Mail: mail servers

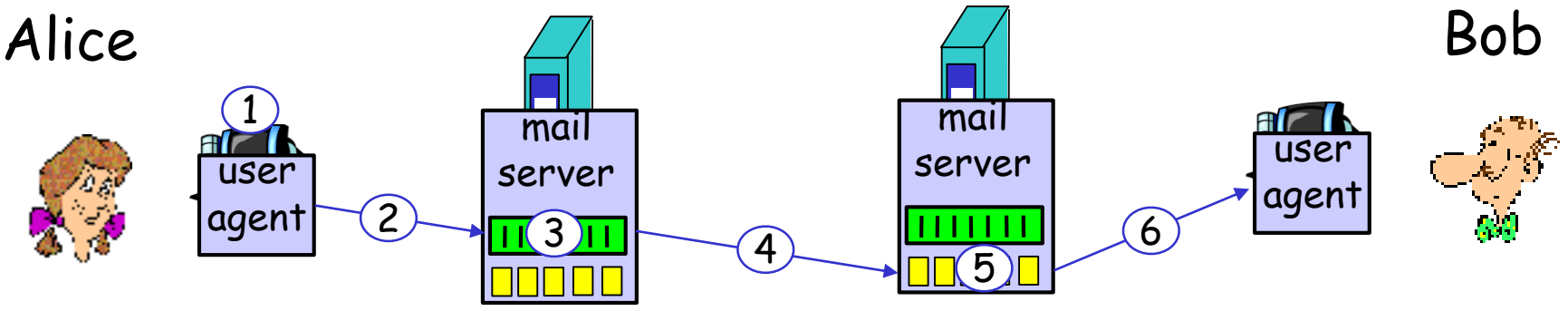
Mail Servers

- mailbox contains incoming messages for user
- message queue of outgoing (to be sent) mail messages
- SMTP protocol between mail servers to send email messages
 - client: sending mail server
 - "server": receiving mail server



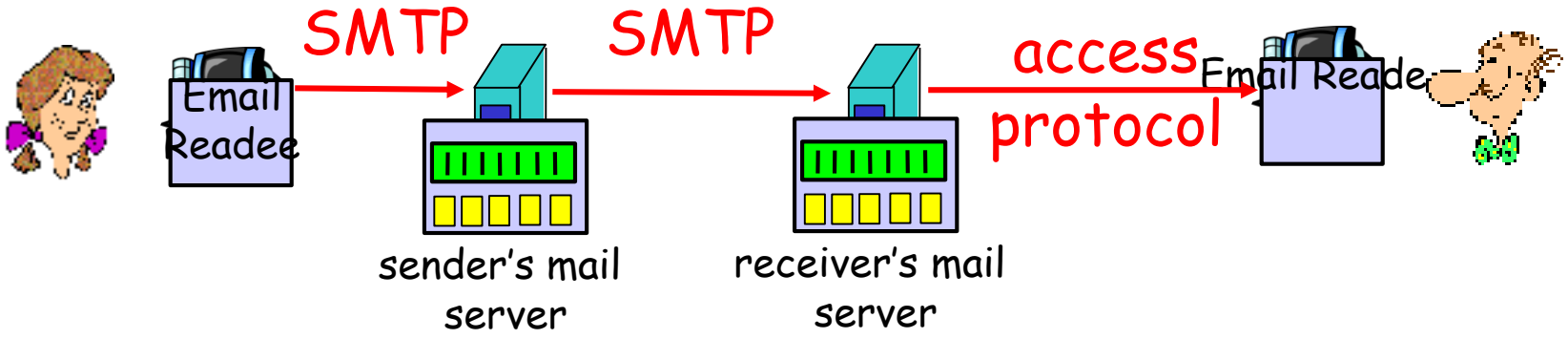
Scenario: Alice sends message to Bob

1) Alice uses Email Reader to compose message and "to" bob@bilkent.edu.tr
2) Alice's UA sends message to her mail server; message placed in message queue
3) Client side of SMTP opens TCP connection with Bob's mail server
4) SMTP client sends Alice's message over the TCP connection
5) Bob's mail server places the message in Bob's mailbox
6) Bob invokes his user agent to read message



Slides adapted from [1]

Mail access protocols



- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
 - POP: Post Office Protocol [RFC 1939]
 - authorization (agent <-->server) and download
 - IMAP: Internet Mail Access Protocol [RFC 1730]
 - more features (more complex)
 - manipulation of stored msgs on server
 - HTTP: Hotmail , Yahoo! Mail, etc.

Electronic Mail: SMTP [RFC 2821]

- uses TCP to reliably transfer email message from client to server, port 25
- direct transfer: sending server to receiving server
- three phases of transfer
 - handshaking (greeting)
 - transfer of messages
 - closure

- Simple Mail Transport Protocol (SMTP) was originally designed for a smaller community of users which was assumed to be well behaved and trust worthy. As such no heed was paid towards incorporating security protocols in it. But with its growth, this trust was breached, owing to lack of adequate security mechanism in it. Several technological and policy changes were made to SMTP servers to make e-mail system secure without creating incompatibility between older and newer systems

How can we add security features to:

- email protocols (SMTP, POP, IMAP)
- Web protocol (HTTP)

- SMTP servers incorporate one or more security features using several add-on e-mail security protocols to make communications secure and private. These protocols use diverse technological means like encryption, symmetric and asymmetric cryptography.
- We will consider two-add-on e-mail security protocols:
- SSL (Secure socket Layer)
- PGP (Pretty Good Privacy).

SSL

- Secure Socket Layer (SSL) and Secure SMTP are encryption based methods that respectively create encrypted secure channel between the sending and receiving MTA's at sockets and transport layers. They are simple methods to obtain e-mail privacy without efforts of the end user.
- But SSL and Secure SMTP provide security only for the path between client and server and not the endpoints that are authenticated by certifying authorities.

What is SSL?

- SSL (Secure Sockets Layer) is a standard security technology for establishing an encrypted link between a server and a client—typically a web server (website) and a browser (Security of websites), email server.
- SSL allows sensitive information such as credit card numbers, social security numbers, and login credentials to be transmitted securely.
- SSL manages server authentication, client authentication and encrypted communication between servers and clients.
- It achieves security in the transport layer.

SSL/TLS

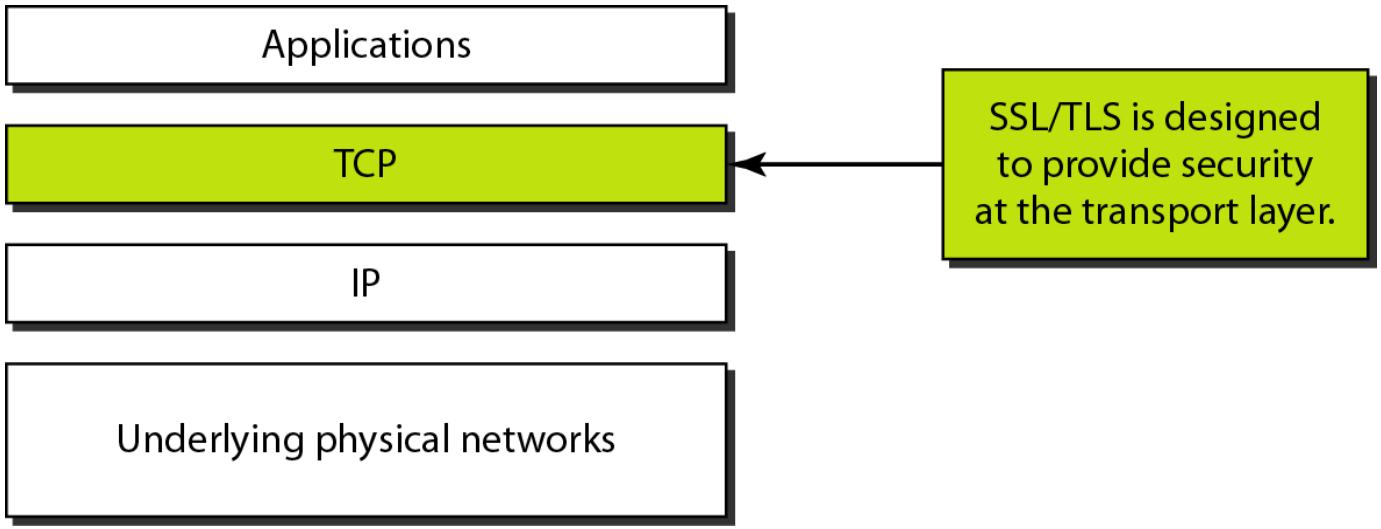
Two protocols are dominant today for providing security at the transport layer: the Secure Sockets Layer (SSL) Protocol and the Transport Layer Security (TLS) Protocol. The latter is actually an IETF version of the former.

- The idea is to provide security services for transactions on the Internet. For example, when a customer shops online, the following security services are desired:
- 1. The customer needs to be sure that the server belongs to the actual vendor, not an imposter. The customer does not want to give an imposter her credit card number (entity authentication). Likewise, the vendor needs to authenticate the customer.
- 2. The customer and the vendor need to be sure that the contents of the message are not modified during transition (message integrity).
- 3. The customer and the vendor need to be sure that an imposter does not intercept sensitive information such as a credit card number (confidentiality).

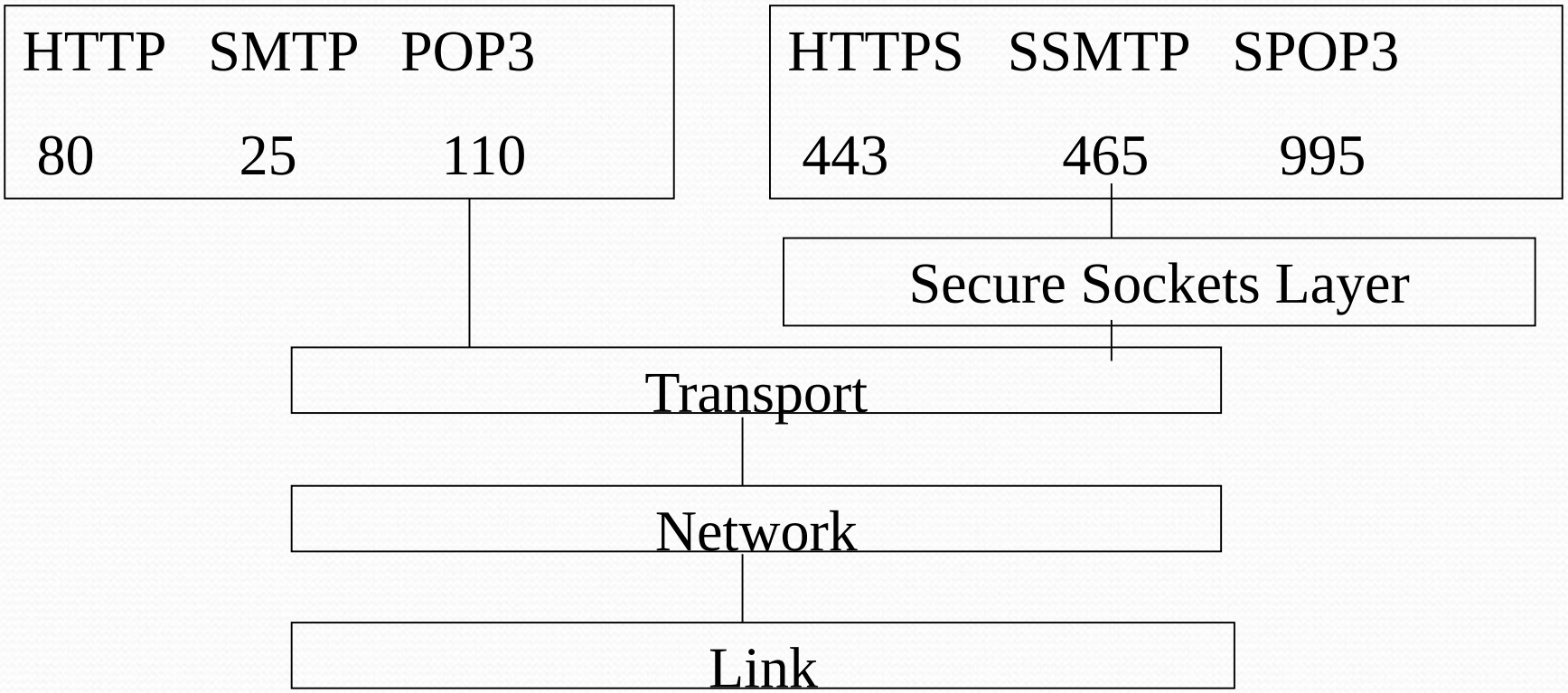
SSL (Secure Socket Layer)

- It is the authentication and encryption mechanism for e-commerce.
- It is originally developed by Netscape
- subsequently became Internet standard known as TLS (Transport Layer Security)
- Secure Socket Layer (SSL) is designed to provide security and compression services to data generated from the application layer. Typically, SSL can receive data from any application layer protocol, but usually the protocol is HTTP. The data received from the application are compressed (optional), signed, and encrypted. The data are then passed to a reliable transport layer protocol such as TCP

Location of SSL and TLS in the Internet model



Where SSL Fits



It operates above the Internet TCP protocol and below high-level application protocols.

What Can SSL Do?

- Allows the server to authenticate itself to the client;
- Allows the client to authenticate itself to the server;
- Allows both the server and the client to establish an encrypted connection.

SSL Services

SSL provides several services on data received from the application layer.

1- Fragmentation:

First, SSL divides the data into blocks of 214 bytes or less.

2- Compression:

Each fragment of data is compressed by using one of the lossless compression methods negotiated between the client and server. This service is optional.

3- Message Integrity:

To preserve the integrity of data, SSL uses a keyed-hash function to create a MAC.

4- Confidentiality:

To provide confidentiality, the original data and the MAC are encrypted using symmetric key cryptography.

5- Framing:

A header is added to the encrypted payload. The is then passed to a reliable transport layer protocol.

Security Parameters

There are cipher suites and cryptographic secrets that together make the security parameters.

- Cipher Suite:

The combination of key exchange, hash, and encryption algorithms defines a cipher suite for each SSL session. Each suite starts with the term SSL, followed by the key-exchange algorithm. The word WITH separates the key exchange algorithm from the encryption and hash algorithms. For example,

SSL_DHE_RSA_WITH_DES_CBC_SHA

defines DHE_RSA (ephemeral Diffie-Hellman with RSA digital signature) as the key exchange with DES_CBC as the encryption algorithm and SHA as the hash algorithm.

SSL chiper suite List

Cipher Suite	*Key Exchange Algorithm*	*Encryption Algorithm*	*Hash Algorithm*
SSL_DHE_RSA_***WITH***_DES_CBC_SHA	DHE_RSA	DES_CBC	SHA
SSL_DHE_RSA_***WITH***_3DES_EDE_CBC_SHA	DHE_RSA	3DES_EDE_CBC	SHA
SSL_DHE_DSS_***WITH***_DES_CBC_SHA	DHE_DSS	DES_CBC	SHA
SSL_DHE_DSS_***WITH***_3DES_EDE_CBC_SHA	DHE_DSS	3DES_EDE_CBC	SHA
SSL_DH_RSA_***WITH***_DES_CBC_SHA	DH_RSA	DES_CBC	SHA
SSL_DH_RSA_***WITH***_3DES_EDE_CBC_SHA	DH_RSA	3DES_EDE_CBC	SHA
SSL_DH_DSS_***WITH***_DES_CBC_SHA	DH_DSS	DES_CBC	SHA
SSL_DH_DSS_***WITH***_3DES_EDE_CBC_SHA	DH_DSS	3DES_EDE_CBC	SHA
SSL_FORTEZZA_DMS_***WITH***_NULL_SHA	FORTEZZA_DMS	NULL	SHA
SSL_FORTEZZA_DMS_***WITH***_FORTEZZA_CBC_SHA	FORTEZZA_DMS	FORTEZZA_CBC	SHA
SSL_FORTEZZA_DMS_***WITH***_RC4_128_SHA	FORTEZZA_DMS	RC4_128	SHA

Cryptographic Secrets

The second part of security parameters is often referred to as cryptographic secrets.

To achieve message integrity and confidentiality, SSL needs six cryptographic secrets, four keys, and two IVs. The client needs one key for message authentication, one key for encryption, and one IV for block encryption. The server needs the same. SSL requires that the keys for one direction be different from those for the other direction. If there is an attack in one direction, the other direction is not affected. These parameters are generated by using a negotiation protocol “handshake protocol”.

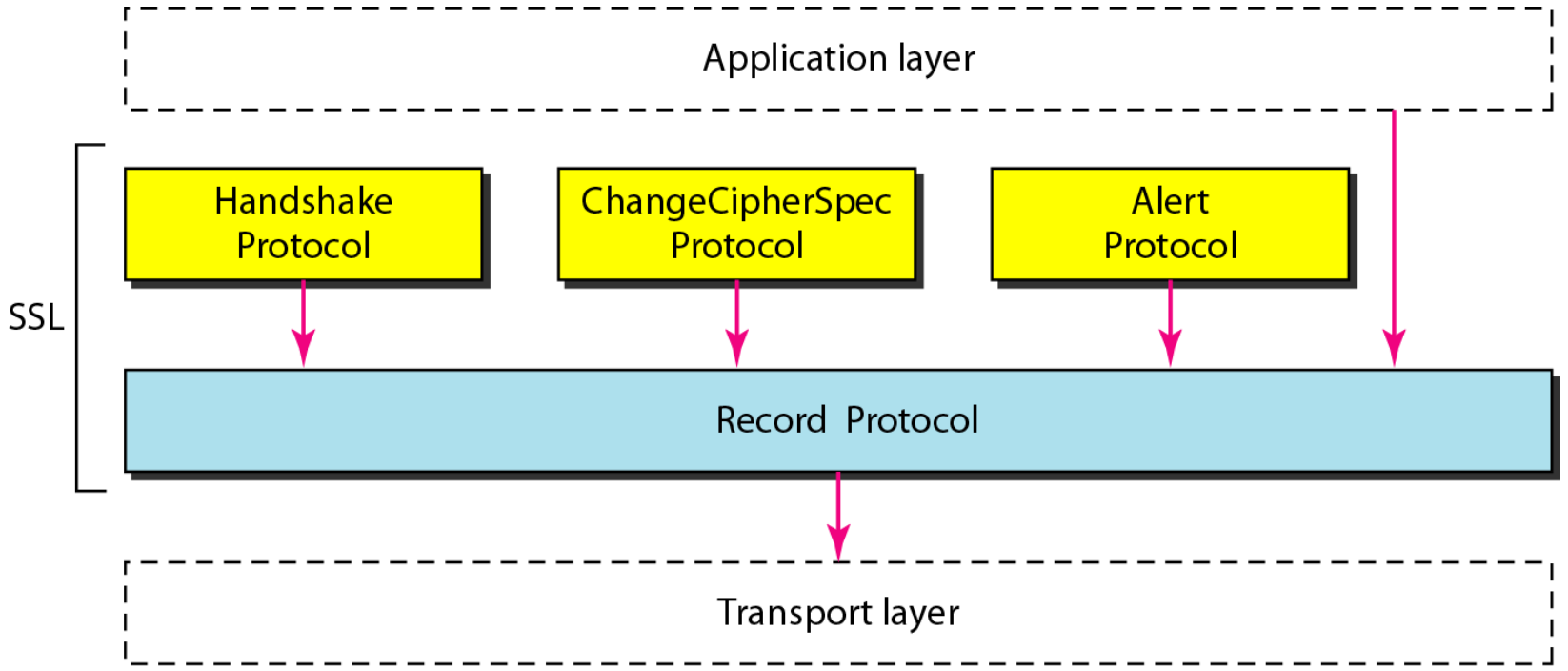
Note

The client and the server have six different cryptography secrets.

SSL Architecture

- **SSL session:**
 - an association between client & server
 - created by the Handshake Protocol
 - define a set of cryptographic parameters
- **SSL connection:**
 - a transient, peer-to-peer, communications link
- A session between two systems is an association that can last for a long time; a connection can be established and broken several times during a session.
- Some of the security parameters are created during the session establishment and are in effect until the session is terminated (for example, cipher suite and master key). Some of the security parameters must be recreated (or occasionally resumed) for each connection (for example, six secrets).

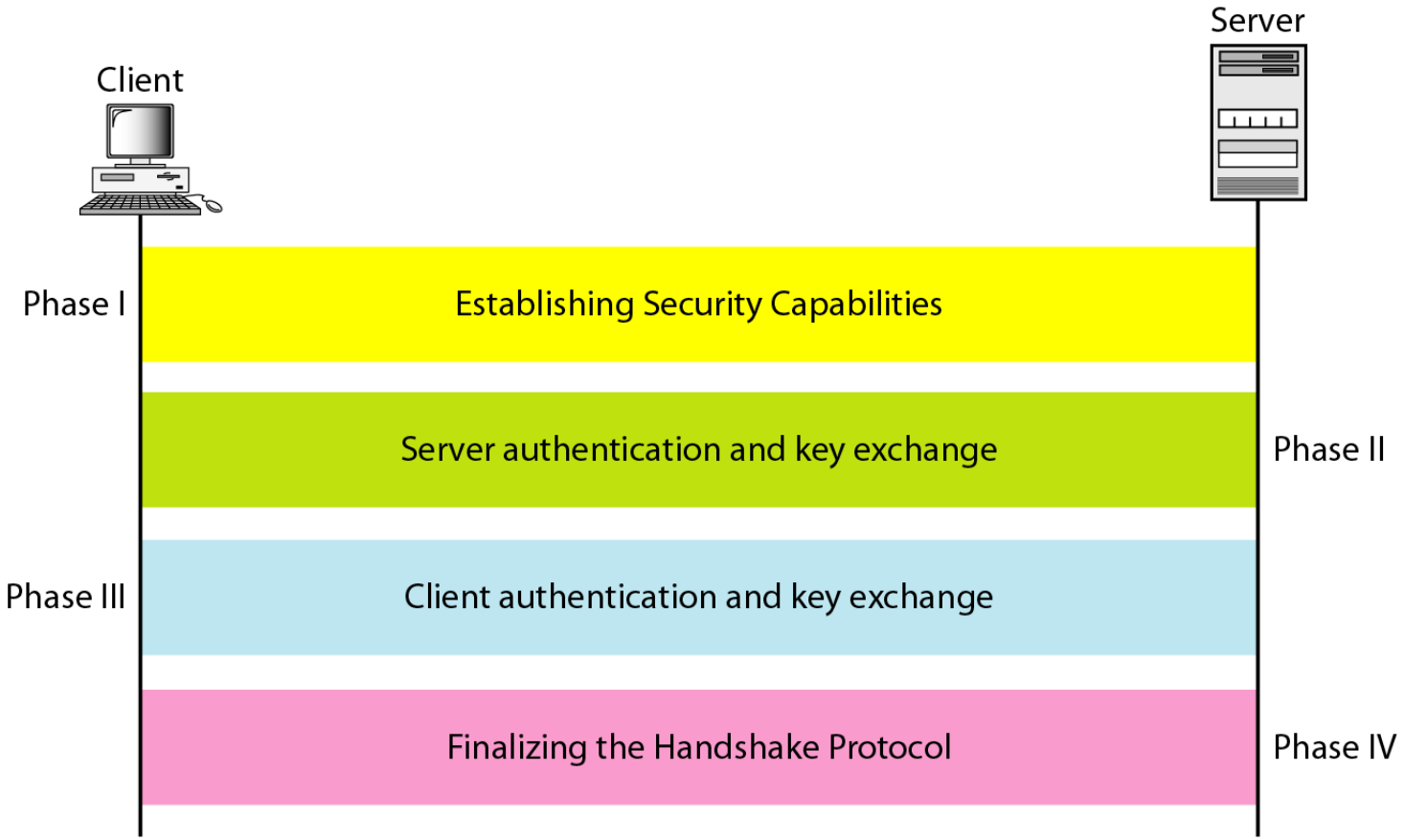
Four SSL protocols



SSL Handshake Protocol

- allows server & client to:
 - authenticate each other
 - to negotiate encryption & MAC algorithms
 - to negotiate cryptographic keys to be used
 - Server Authentication and Key Exchange
 - Client Authentication and Key Exchange
 - Finish

Figure *Handshake Protocol*



SSL defines four protocols in two layers. The Record Protocol is the carrier. It carries messages from three other protocols as well as the data coming from the application layer. Messages from the Record Protocol are payloads to the transport layer, normally TCP. The Handshake Protocol provides security parameters for the Record Protocol. It establishes a cipher set and provides keys and security parameters. It also authenticates the server to the client and the client to the server, if needed. The ChangeCipherSpec Protocol is used for signaling the readiness of cryptographic secrets. The Alert Protocol is used to report abnormal conditions.

ChangeCipherSpec Protocol:

We have seen that the negotiation of the cipher suite and the generation of crypto graphic secrets are formed gradually during the Handshake Protocol. The question now is, When can the two parties use these parameter secrets? SSL mandates that the parties not use these parameters or secrets until they have sent or received a special message, the ChangeCipherSpec message, which is exchanged during the Handshake Protocol and defined in the ChangeCipherSpec Protocol. Before the exchange of any ChangeCipherSpec messages, only the pending columns have values.

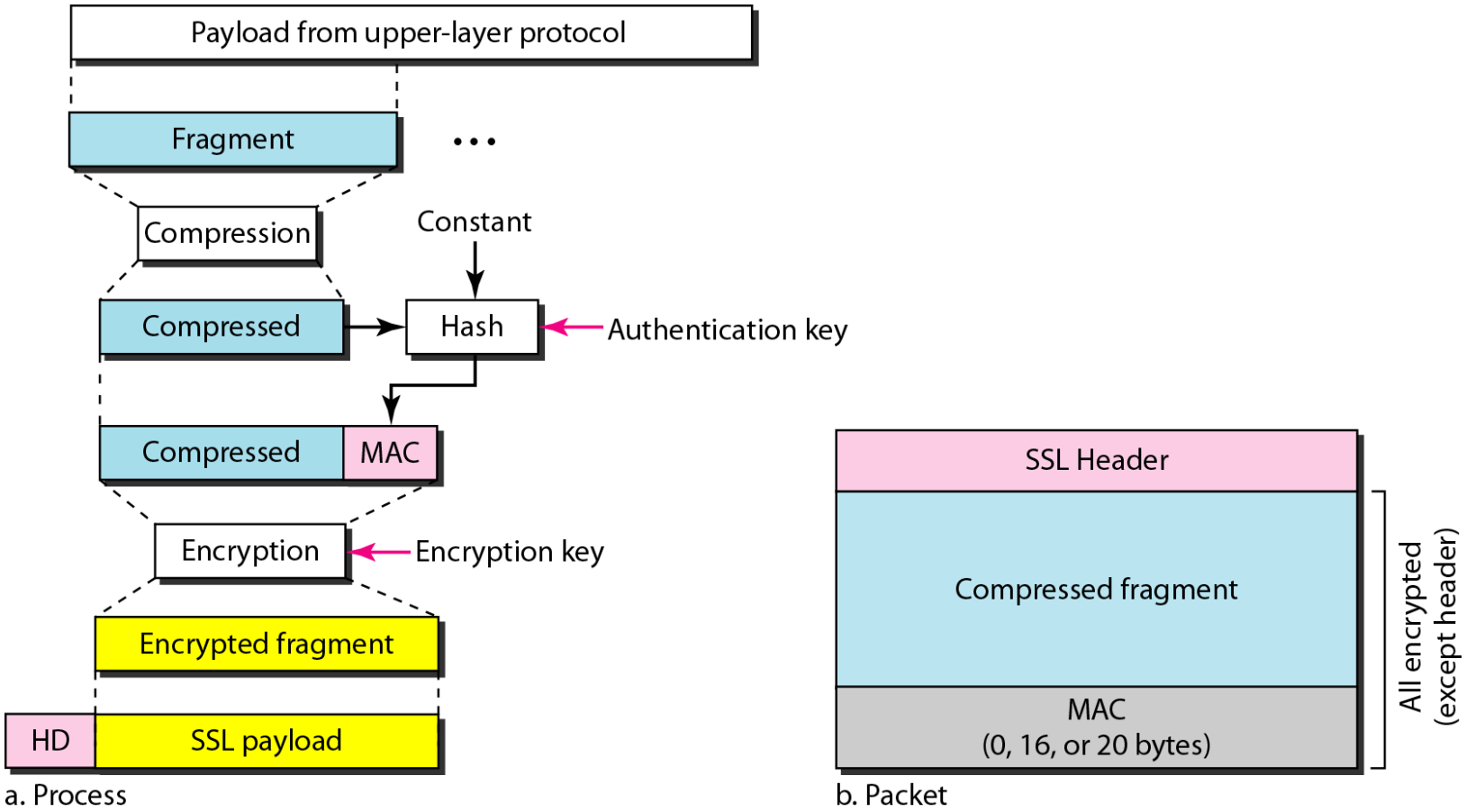
- Alert Protocol:

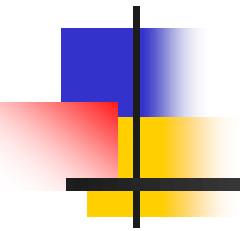
SSL uses the Alert Protocol for reporting errors and abnormal conditions. It has only one message type, the alert message that describes the problem and its level (warning or fatal).

- Record Protocol:

The Record Protocol carries messages from the upper layer (Handshake Protocol, ChangeCipherSpec Protocol, Alert Protocol, or application layer). The message is fragmented and optionally compressed; a MAC is added to the compressed message by using the negotiated hash algorithm. The compressed fragment and the MAC are encrypted by using the negotiated encryption algorithm. Finally, the SSL header is added to the encrypted message..

Figure *Processing done by the Record Protocol*





E-mail Security

©The McGraw-Hill Companies, Inc., 2000

E-Mail Security

- A number of cryptosystems have been adapted to help secure e-mail:
- Secure Multipurpose Internet Mail Extensions (SMIME): add encryption and authentication via digital signatures based on public key cryptosystems.
- Privacy Enhanced Mail (PEM): has been proposed by the Internet Engineering Task Force (IETF) , it uses 3DES symmetric key encryption and RSA for digital signature.
- Pretty Good Privacy (PGP): use IDEA (the International Data Encryption Algorithm is a replacement for the (DES)) cipher.

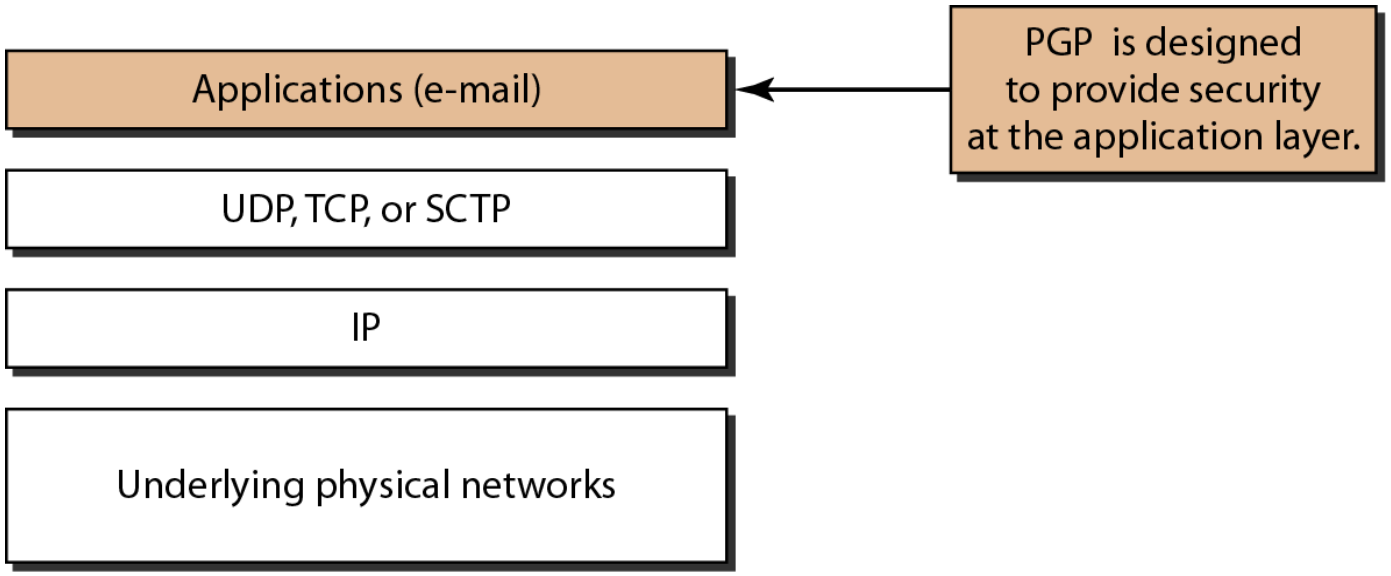
PGP

One of the protocols to provide security at the application layer is Pretty Good Privacy (PGP). PGP is designed to create authenticated and confidential e-mails.

Topics discussed in this section:

Security Parameters
Services
A Scenario
PGP Algorithms
Key Rings
PGP Certificates

Figure *Position of PGP in the TCP/IP protocol suite*



Pretty Good Privacy (PGP)

- Pretty Good Privacy (PGP) is a widely used approach to providing security for electronic mail. It provides authentication, confidentiality, data integrity, and non-repudiation.
- Originally devised by Phil Zimmerman, it has evolved into an IETF standard known as OpenPGP.

- It is a hybrid protocol .
- It uses both symmetric key and public key encryption: RSA and DSS (Digital Signature Standard) for public key certificates.
- It uses IDEA (the International Data Encryption Algorithm) which is a replacement for the (DES)) cipher.

- PGP combines IDEA and RSA:
 - IDEA fast, but symmetric, hence key distribution problem
 - RSA slower, but no key distribution problem
 - Solution: Use RSA to encrypt and distribute key for IDEA encryption!!!

- Sending an e-mail is a one-time activity. The nature of this activity is different from those we have seen in the SSL. SSL, we assume that the two parties create a session between themselves and exchange data in both directions. In e-mail, there is no session. Alice and Bob cannot create a session. Alice sends a message to Bob; sometime later, Bob reads the message and mayor may not send a reply. We discuss the security of a unidirectional message because what Alice sends to Bob is totally independent of what Bob sends to Alice.

PGP Services

PGP can provide several services based on the requirements of the user. An e-mail can use one or more of these services.

1-Plaintext:

The simplest case is to send the e-mail message in plaintext (no service). Alice, the sender, composes a message and sends it to Bob, the receiver. The message is stored in Bob's mailbox until it is retrieved by him.

2-Message Authentication:

Probably the next improvement is to let Alice sign the message. Alice creates a digest of the message and signs it with her private key.

When Bob receives the message, he verifies the message by using Alice's public key. Two keys are needed for this scenario. Alice needs to know her private key; Bob needs to know Alice's public key.

3- Compression:

A further improvement is to compress the message and digest to make the packet more compact. This improvement has no security benefit, but it eases the traffic.

4-Confidentiality with One Time Session Key:

As we discussed before, confidentiality in an e-mail system can be achieved by using conventional encryption with a one-time session key.

Alice can create a session key, use the session key to encrypt the message and the digest, and send the key itself with the message. However, to protect the session key, Alice encrypts it with Bob's public key.

5-Code Conversion:

Another service provided by PGP is code conversion. Most systems allow the message to consist of only ASCII characters. To translate other characters not in the ASCII set, PGP uses Radix 64 conversion. Each character to be sent (after encryption) is converted to Radix 64 code.

5-Segmentation:

PGP allows segmentation of the message after it has been converted to Radix 64 to make each transmitted unit the uniform size allowed by the underlying e-mail protocol.

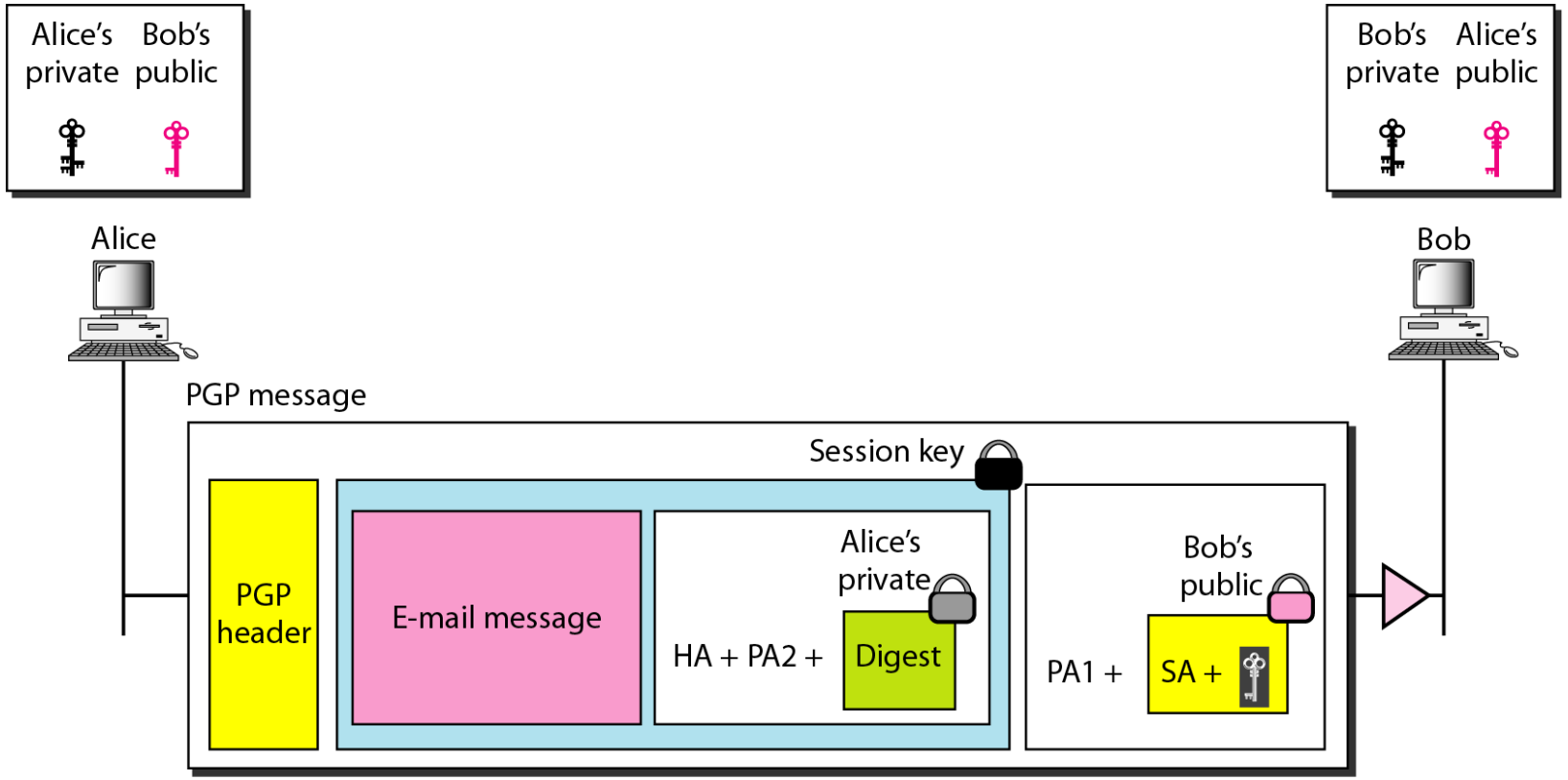
A Scenario for communication

- Let us describe a scenario that combines some of these services, authentication and confidentiality.
- The whole idea of PGP is based on the assumption that a group of people who need to exchange messages trust one another. Everyone in the group somehow knows (with a degree of trust) the public key of any other person in the group. Based on this single assumption, the following Figure shows a simple scenario in which an authenticated and encrypted message is sent from Alice to Bob.

Note

In PGP, the sender of the message needs to include the identifiers of the algorithms used in the message as well as the values of the keys.

Figure *A scenario in which an e-mail message is authenticated and encrypted*

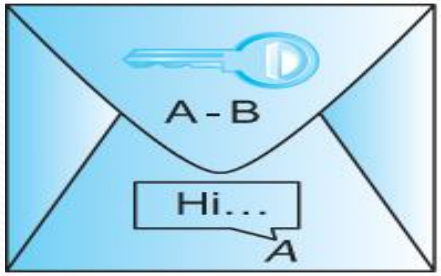


PA1: Public-key algorithm 1 (for encrypting session key)
PA2: Public-key algorithm (for encrypting the digest)
SA: Symmetric-key algorithm identification (for encrypting message and digest)
HA: Hash algorithm identification (for creating digest)

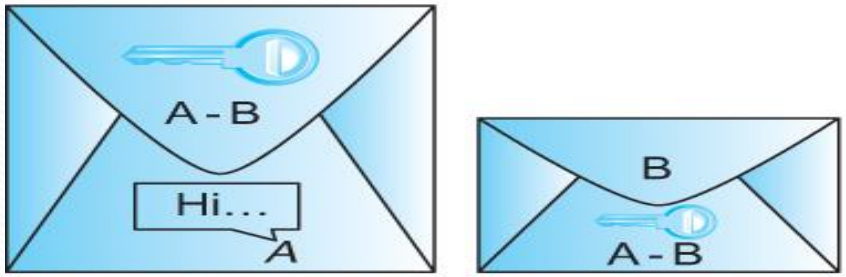
Hi... = The plaintext message



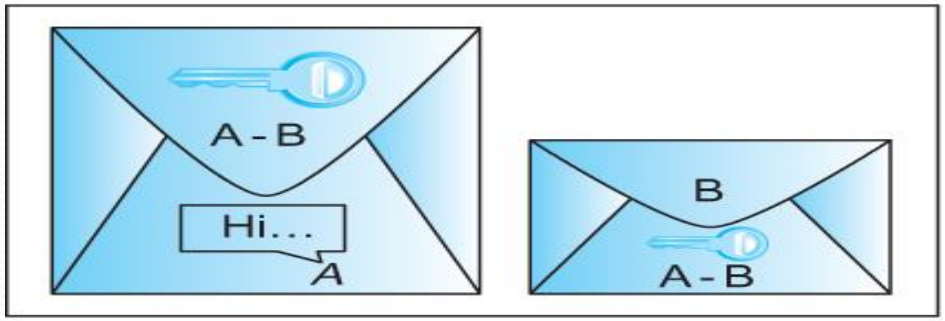
1) Digitally sign using Alice's private key



2) Encrypt using a newly generated one-time session key



3) Encrypt the session key using Bob's public key, and append that



4) Use base64 encoding to obtain an ASCII-compatible representation

- Sender Site:

The following shows the steps used in this scenario at Alice's site:

1. Alice creates a session key (for symmetric encryption/decryption) and concatenates it with the identity of the algorithm which will use this key. The result is encrypted with Bob's public key. Alice adds the identification of the public-key algorithm used above to the encrypted result. This part of the message contains three pieces of information: the session key, the symmetric encryption/decryption algorithm to be used later, and the asymmetric encryption/decryption algorithm that was used for this part.

- 2.

a. Alice authenticates the message (e-mail) by using a public-key signature algorithm and encrypts it with her private key. The result is called the signature.

Alice appends the identification of the public key (used for encryption) as well as the identification of the hash algorithm (used for authentication) to the signature. This part of the message contains the signature and two extra pieces of information: the encryption algorithm and the hash algorithm.

b. Alice concatenates the three pieces of information created above with the message (e-mail) and encrypts the whole thing, using the session key created in step 1.

3. Alice combines the results of steps 1 and 2 and sends them to Bob (after adding true appropriate PGP header).

- Receiver Site:

The following shows the steps used in this scenario at Bob's side after he has received the PGP packet:

1. Bob uses his private key to decrypt the combination of the session key and symmetric-key algorithm identification.

2. Bob uses the session key and the algorithm obtained in step 1 to decrypt the rest of the PGP message. Bob now has the content of the message, the identification of the public algorithm used for creating and encrypting the signature, and the identification of the hash algorithm used to create the hash out of the message.

3. Bob uses Alice's public key and the algorithm defined by PA2 to decrypt the digest.

4. Bob uses the hash algorithm defined by HA to create a hash out of message he obtained in step 2.

5. Bob compares the hash created in step 4 and the hash he decrypted in step 3. If the two are identical, he accepts the message; otherwise, he discards the message.

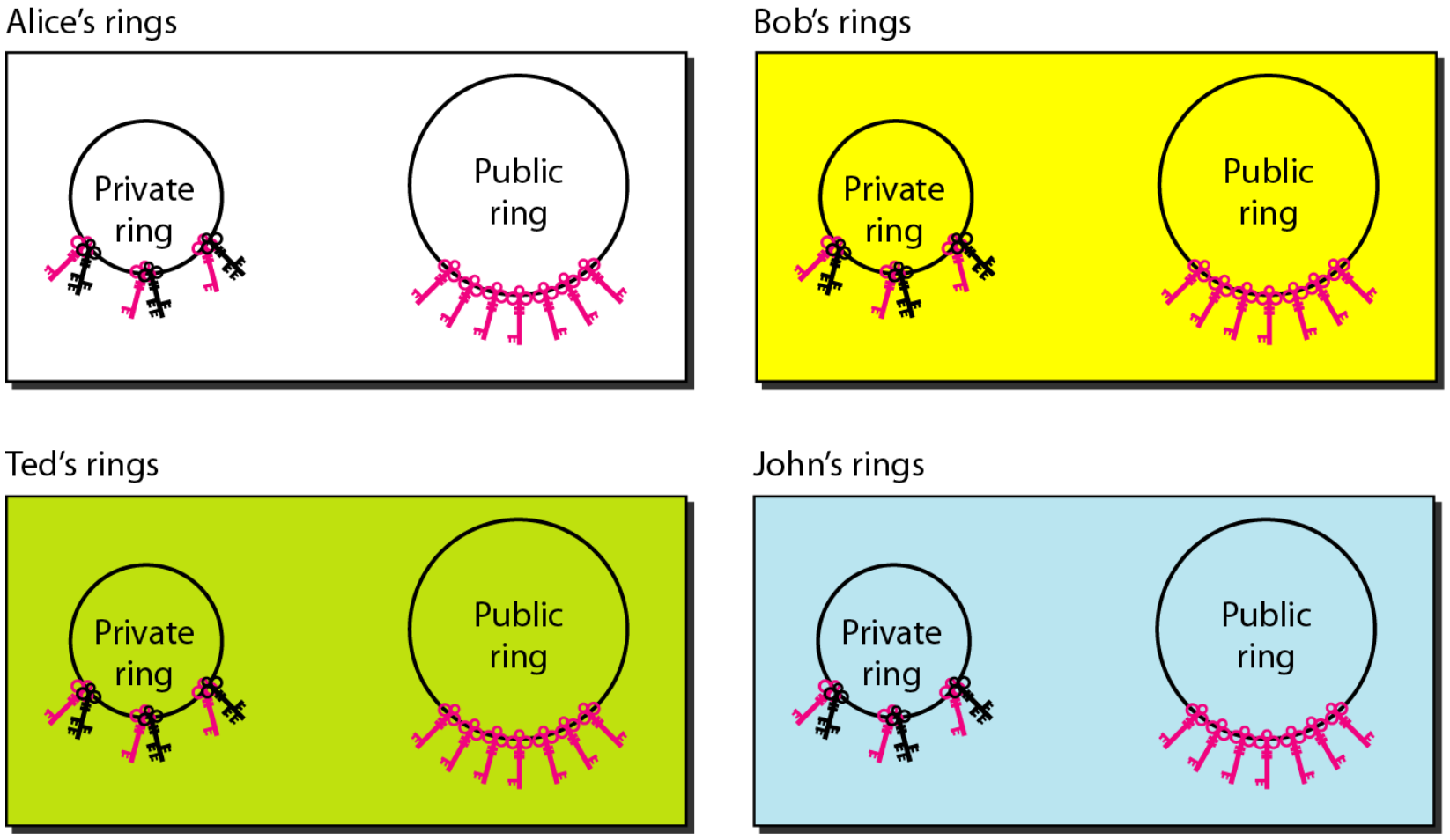
Table *PGP Algorithms*

Algorithm	*ID*	*Description*
Public key	1	RSA (encryption or signing)
	2	RSA (for encryption only)
	3	RSA (for signing only)
	17	DSS (for signing)
Hash algorithm	1	MD5
	2	SHA-1
	3	RIPE-MD
Encryption	0	No encryption
	1	IDEA
	2	Triple DES
	9	AES

Key Rings

In the previous scenarios, we assumed that Alice needed to send a message to only Bob. That is not always the case. Alice may need to send messages to many people. In this case, Alice needs a key ring of public keys, with a key belonging to each person with whom Alice needs to correspond (send or receive messages). In addition, the PGP designers specified a ring of private/public keys. One reason is that Alice may wish to change her pair of keys from time to time. Another reason is that Alice may need to correspond with different groups of people (friends, colleagues, and so on). Alice may wish to use a different key pair for each group. Therefore, each user needs to have two sets of rings: a ring of private/public keys and a ring of public keys of other people. The following shows a community of four people, each having a ring of pairs of private/public keys and, at the same time, a ring of four public keys belonging to the other four people in the community. The figure shows seven public keys for each public ring. Each person in the ring can keep more than one public key for each other person.

Figure *Rings*



Alice, for example, has several pairs of private/public keys belonging to her and public keys belonging to other people. Note that everyone can have more than one public key. Two cases may arise.

1. Alice needs to send a message to one of the persons in the community.

a. She uses her private key to sign the digest.

b. She uses the receiver's public key to encrypt a newly created session key.

c. She encrypts the message and signs the digest with the session key created.

2. Alice receives a message from one of the persons in the community.

a. She uses her private key to decrypt the session key.

b. She uses the session key to decrypt the message and digest.

c. She uses her public key to verify the digest.

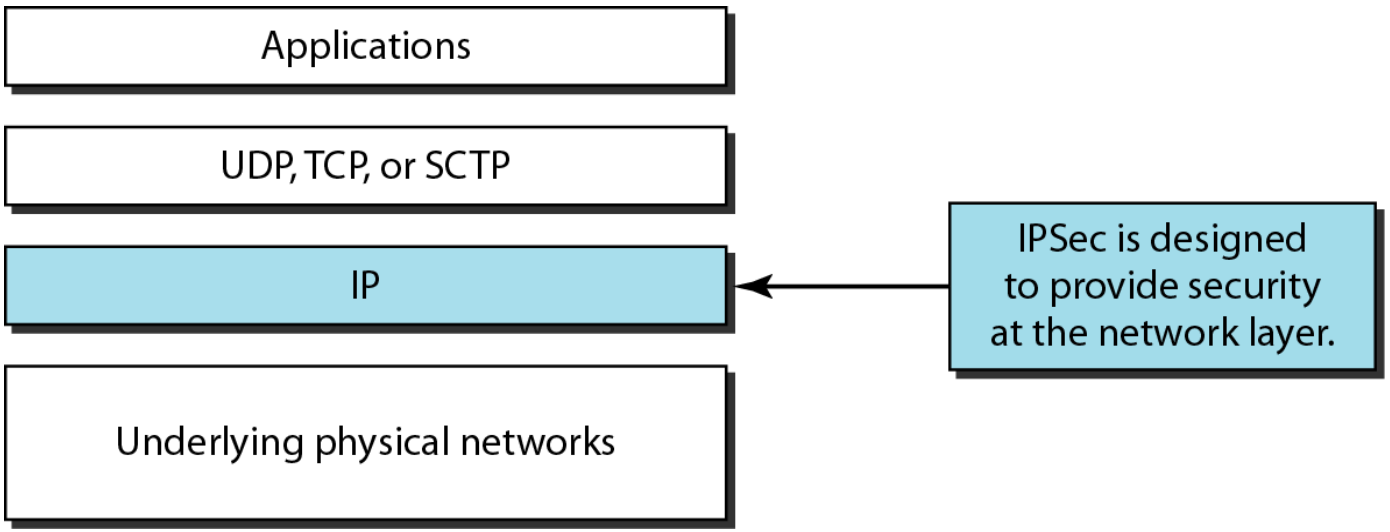
IPSecurity (IPSec)

IPSecurity (IPSec) is a collection of protocols designed by the Internet Engineering Task Force (IETF) to provide security for a packet at the network level.

Topics discussed in this section:

Two Modes
Two Security Protocols
Security Association
Internet Key Exchange (IKE)
Virtual Private Network

Figure *TCP/IP protocol suite and IPSec*



Two Modes:

IPSec operates in one of two different modes: the transport mode or the tunnel mode.

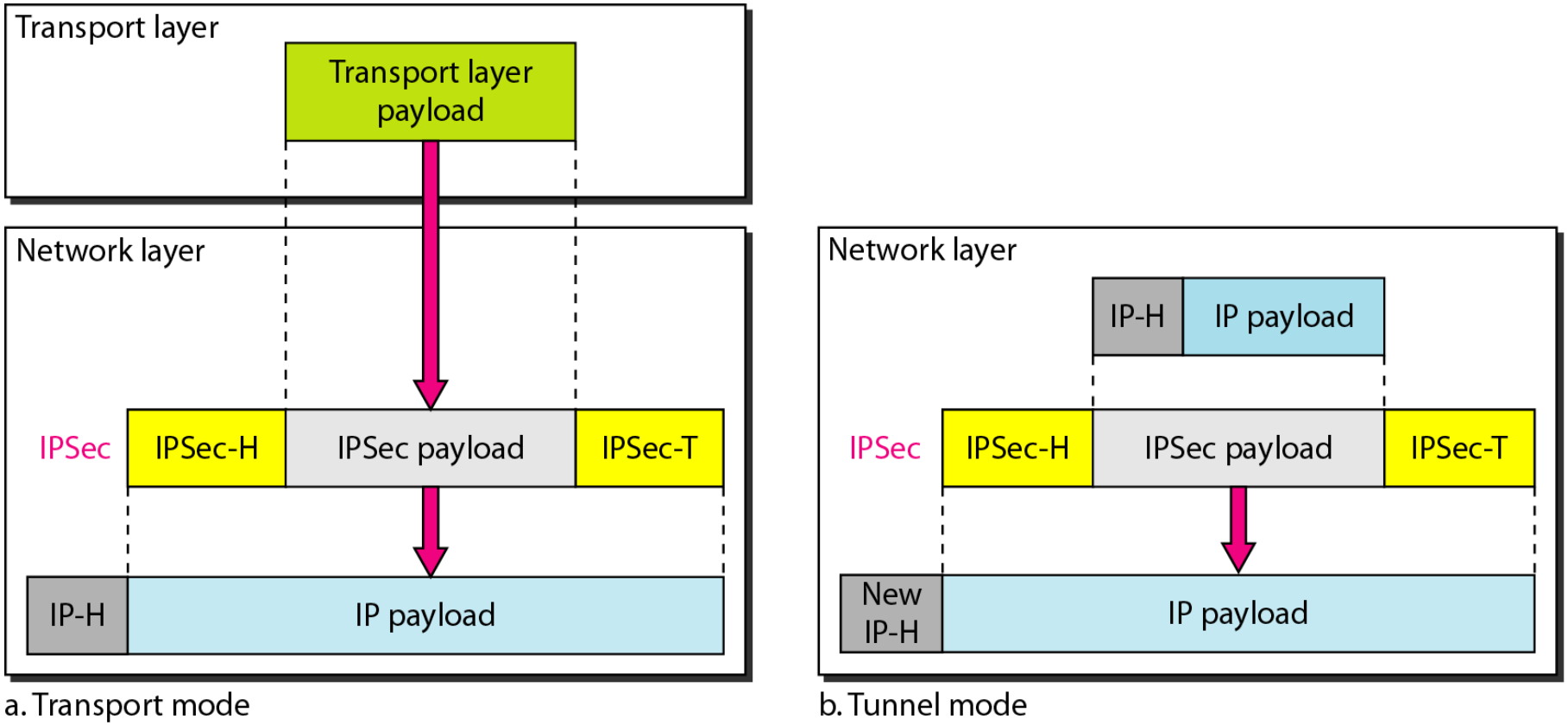
1. Transport Mode:

In the transport mode, IPSec protects what is delivered from the transport layer to the network layer. In other words, the transport mode protects the network layer payload, the payload to be encapsulated in the network layer.

Note that the transport mode does not protect the IP header. In other words, the transport mode does not protect the whole IP packet; it protects only the packet from the transport layer (the IP layer payload). In this mode, the IPSec header and trailer are added to the information corning from the transport layer. The IP header is added later.

The transport mode is normally used when we need host-to-host (end-to-end) protection of data. The sending host uses IPSec to authenticate and/or encrypt the payload delivered from the transport layer. The receiving host uses IPSec to check the authentication and/or decrypt the IP packet and deliver it to the transport layer.

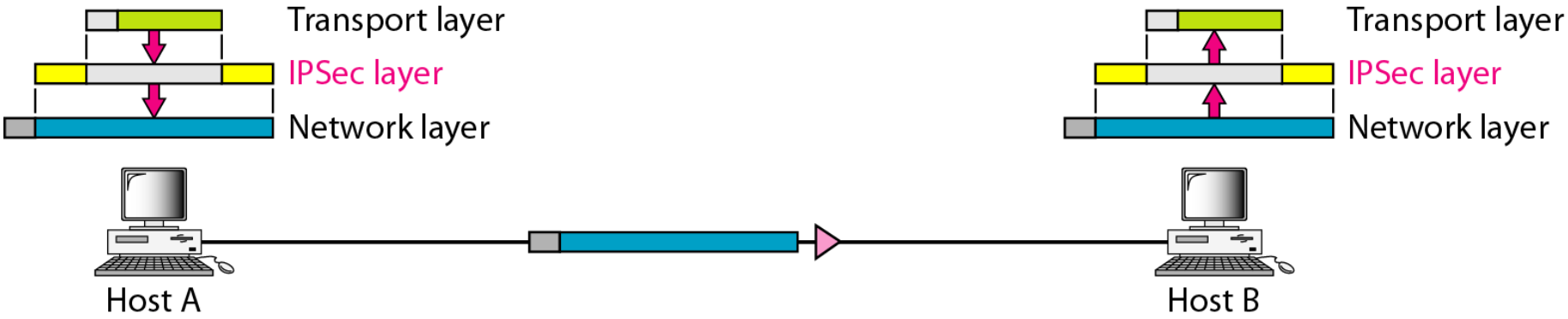
Figure *Transport mode and tunnel modes of IPSec protocol*



Note

IPSec in the transport mode does not protect the IP header; it only protects the information coming from the transport layer.

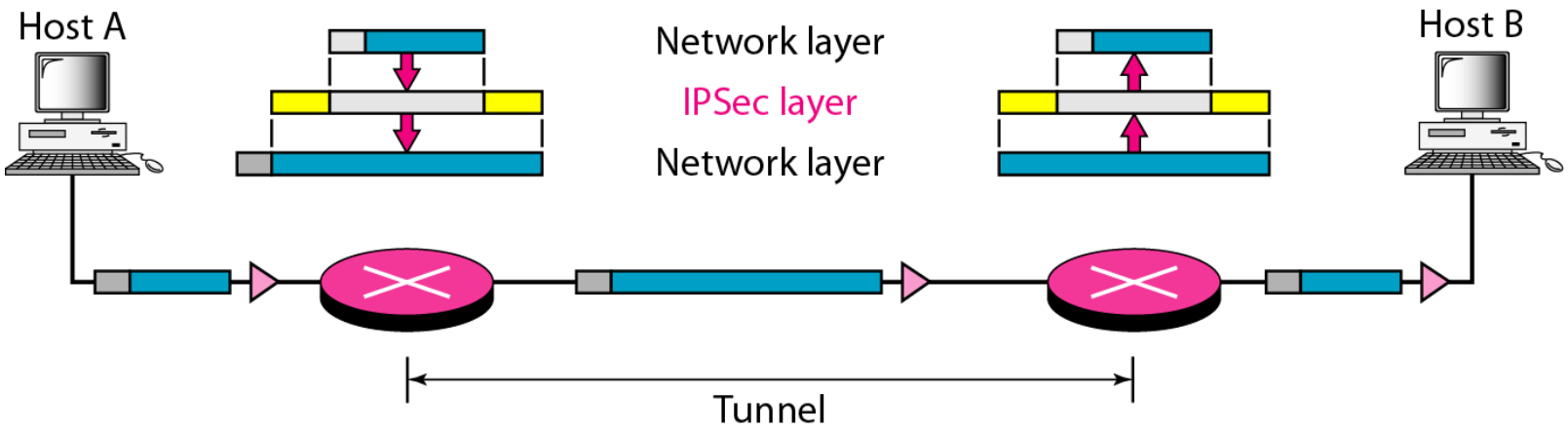
Figure *Transport mode in action*



2- Tunnel Mode:

In the tunnel mode, IPSec protects the entire IP packet. It takes an IP packet, including the header, applies IPSec security methods to the entire packet, and then adds a new IP header .The new IP header, as we will see shortly, has different information than the original IF header. The tunnel mode is normally used between two routers, between a host and a router, or between a router and a host. In other words, we use the tunnel mode when either the sender or the receiver is not a host. The entire original packet is protected from intrusion between the sender and the receiver. It's as if the whole packet goes through an imaginary tunnel.

Figure *Tunnel mode in action*



Note

IPSec in tunnel mode protects the original IP header.

Two Security Protocols:

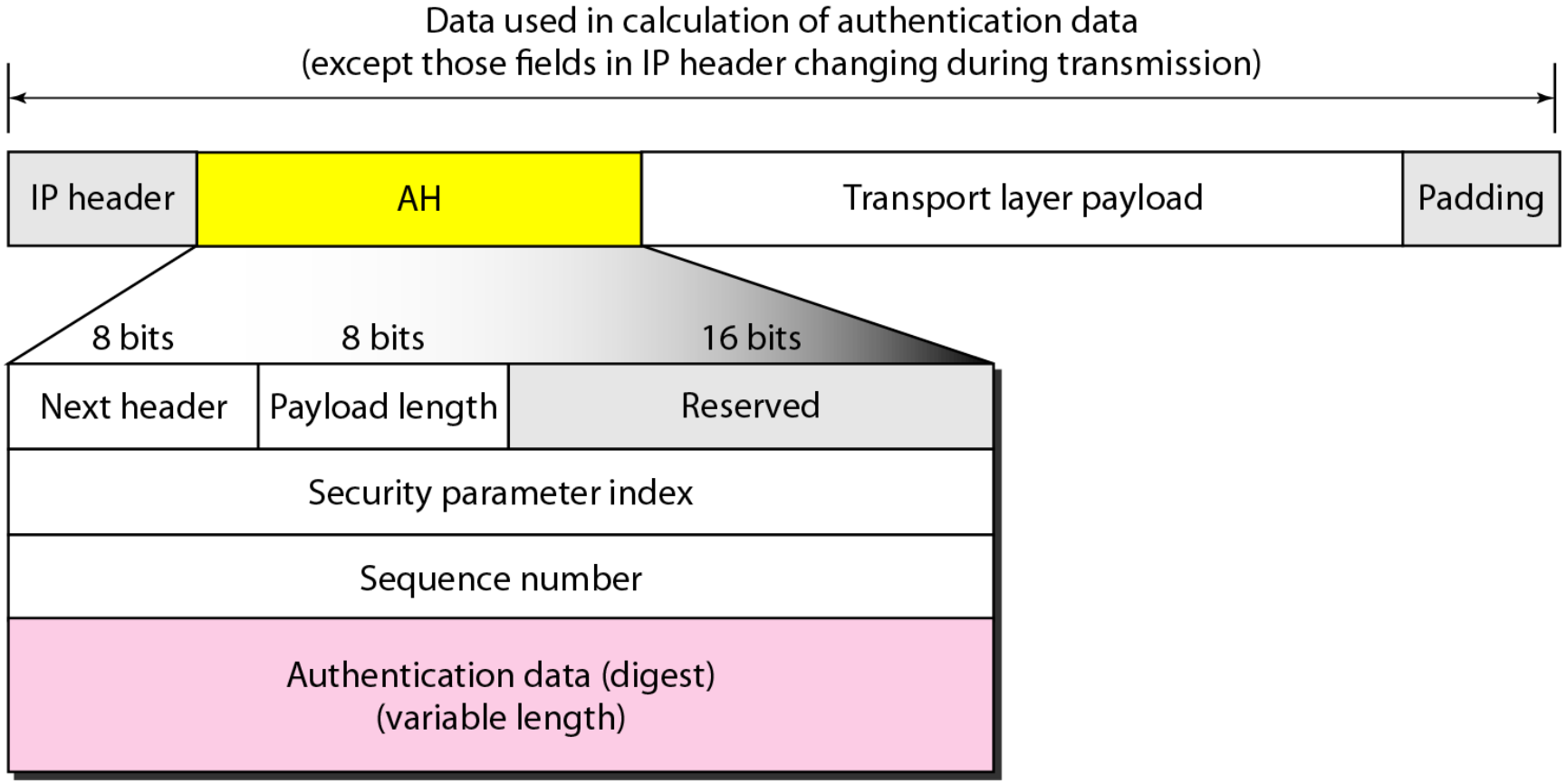
IPSec defines two protocols-the Authentication Header (AH) Protocol and the Encapsulating Security Payload (ESP) Protocol-to provide authentication and/or encryption for packets at the IP level.

1- Authentication Header (AH):

The Authentication Header (AH) Protocol is designed to authenticate the source host and to ensure the integrity of the payload carried in the IP packet. The protocol uses a hash function and a symmetric key to create a message digest; the digest is inserted in the authentication header. The AH is then placed in the appropriate location based on the mode (transport or tunnel.

When an IP datagram carries an authentication header, the original value in the protocol field of the IP header is replaced by the value 51. A field inside the authentication header (the next header field) holds the original value of the protocol field (the type of payload being carried by the IP datagram).

Figure *Authentication Header (AH) Protocol in transport mode*



The addition of an authentication header follows these steps:

1. An authentication header is added to the payload with the authentication data field set to zero.
2. Padding may be added to make the total length even for a particular hashing algorithm.
3. Hashing is based on the total packet. However, only those fields of the IP header that do not change during transmission are included in the calculation of the message digest (authentication data).

4. The authentication data are inserted in the authentication header.

5. The IP header is added after the value of the protocol field is changed to 51.

A brief description of each field follows:

1.Next header: The 8-bit next-header field defines the type of payload carried by the IP datagram (such as TCP, UDP). It has the same function as the protocol field in the IP header before encapsulation

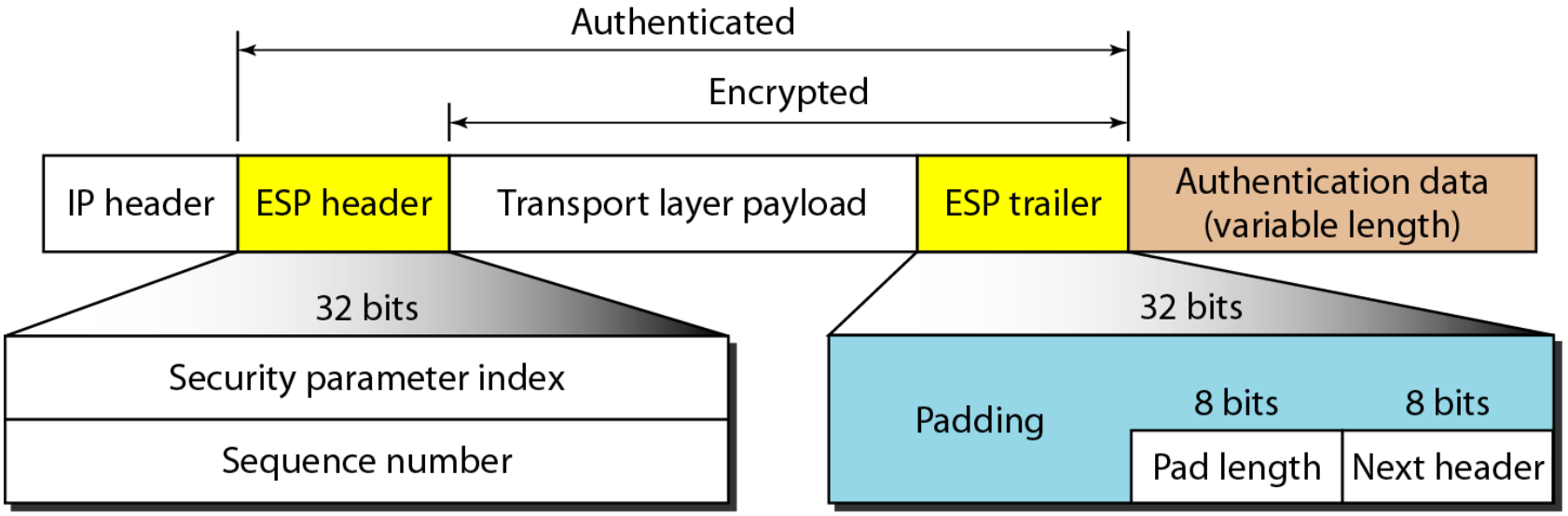
2. Payload length:
The name of this 8-bit field is misleading. It does not define the length of the payload; it defines the length of the authentication header in 4-byte multiples, but it does not include the first 8 bytes.

3.Security parameter index:
The 32-bit security parameter index (SPI) field plays
the role of a virtual-circuit identifier and is the same for all packets sent during a connection called a security association (discussed later).

4. Sequence number:
A 32-bit sequence number provides ordering information for a sequence of data-grams. The sequence numbers prevent a playback. Note that the sequence number is not repeated even if a packet is retransmitted. A sequence number does not wrap around after it reaches $2^{32;}$ a new connection must be established.

5. Authentication data
Finally, the authentication data field is the result of applying a hash function to the entire IP datagram except for the fields that are changed during transit (e.g., time-to-live),

Figure *Encapsulating Security Payload (ESP) Protocol in transport mode*



Encapsulating Security Payload (ESP):

The AH Protocol does not provide privacy, only source authentication and data integrity. IPSec later defined an alternative protocol that provides source authentication, integrity, and privacy called Encapsulating Security Payload (ESP). ESP adds a header and trailer. Note that ESP's authentication data are added at the end of the packet which makes its calculation easier. When an IP datagram carries an ESP header and trailer, the value of the protocol field in the IP header is 50. A field inside the ESP trailer (the next-header field) holds the original value of the protocol field (the type of payload being carried by the IP datagram, such as TCP or UDP).

The ESP procedure follows these steps:

1. An ESP trailer is added to the payload.
2. The payload and the trailer are encrypted.
3. The ESP header is added.
4. The ESP header, payload, and ESP trailer are used to create the authentication data.
5. The authentication data are added to the end of the ESP trailer.
6. The IP header is added after the protocol value is changed to 50.

The fields for the header and trailer are as follows:

1. Security parameter index:

The 32-bit security parameter index field is similar to that defined for the AH Protocol.

Note

ESP provides source authentication, data integrity, and privacy.

2. Sequence number:
The 32-bit sequence number field is similar to that defined for the AH Protocol.

3. Padding:
This variable-length field (0 to 255 bytes) of Os serves as padding.

4. Pad length:
The 8-bitpadlength field defines the number of padding bytes. The value is between 0 and 255; the maximum value is rare.

5. Next header:
The 8-bitnext-header field is similar to that defined in the AH Protocol. It serves the same purpose as the protocol field in the IP header before encapsulation.

6. Authentication data:
Finally, the authentication data field is the result of applying an authentication scheme to parts of the datagram. Note the difference between the authentication data in AH and ESP. In AH, part of the IP header is included in the calculation of the authentication data; in ESP, it is not.

Table *IPSec services*

Services	*AH*	*ESP*
Access control	Yes	Yes
Message authentication (message integrity)	Yes	Yes
Entity authentication (data source authentication)	Yes	Yes
Confidentiality	No	Yes
Replay attack protection	Yes	Yes

IPv4 and IPv6:

IPSec supports both IPv4 and IPv6. In IPv6, however, AH and ESP are part of the extension header.

- AH Versus ESP:

The ESP Protocol was designed after the AH Protocol was already in use. ESP does whatever AH does with additional functionality (privacy). The question is, Why do we need AH? The answer is, We don't. However, the implementation of AH is already included in some commercial products, which means that AH will remain part of the Internet until the products are phased out.

Virtual Private Network

Virtual private network (VPN) is a technology that is gaining popularity among large organizations that use the global Internet for both intra- and interorganizational communication, but require privacy in their internal communications. We discuss VPN here because it uses the IPSec Protocol to apply security to the IP datagrams.

- Private Networks:

A private network is designed for use inside an organization. It allows access to shared resources and, at the same time, provides privacy. Before we discuss some aspects of these networks, let us define two commonly used, related terms: intranet and extranet.

Intranet An intranet is a private network (LAN) that uses the Internet model. However, access to the network is limited to the users inside the organization. The network uses application programs defined for the global Internet, such as HTTP, and may have Web servers, print servers, file servers, and so on.

Extranet An extranet is the same as an intranet with one major difference: Some resources may be accessed by specific groups of users outside the organization under the control of the network administrator. For example, an organization may allow authorized customers access to product specifications, availability, and online ordering. A university or a college can allow distance learning students access to the computer lab after passwords have been checked.

Addressing A private network that uses the Internet model must use IP addresses. Three choices are available:

1. The network can apply for a set of addresses from the Internet authorities and use them without being connected to the Internet. This strategy has an advantage. If in the future the organization decides to be connected to the Internet, it can do so with relative ease. However, there is also a disadvantage: The address space is wasted in the meantime.

2. The network can use any set of addresses without registering with the Internet authorities. Because the network is isolated, the addresses do not have to be unique. However, this strategy has a serious drawback: Users might mistakenly confuse the addresses as part of the global Internet.

3. To overcome the problems associated with the first and second strategies, the Internet authorities have reserved three sets of addresses, shown in the Table Any organization can use an address out of this set without permission from the Internet authorities. Everybody knows that these reserved addresses are for private networks. They are unique inside the organization, but they are not unique globally. No router will forward a packet that has one of these addresses as the destination address.

Table 32.2 ***Addresses for private networks***

Prefix	*Range*	*Total*
10/8	10.0.0.0 to 10.255.255.255	2^{24}
172.16/12	172.16.0.0 to 172.31.255.255	2^{20}
192.168/16	192.168.0.0 to 192.168.255.255	2^{16}

Achieving Privacy:

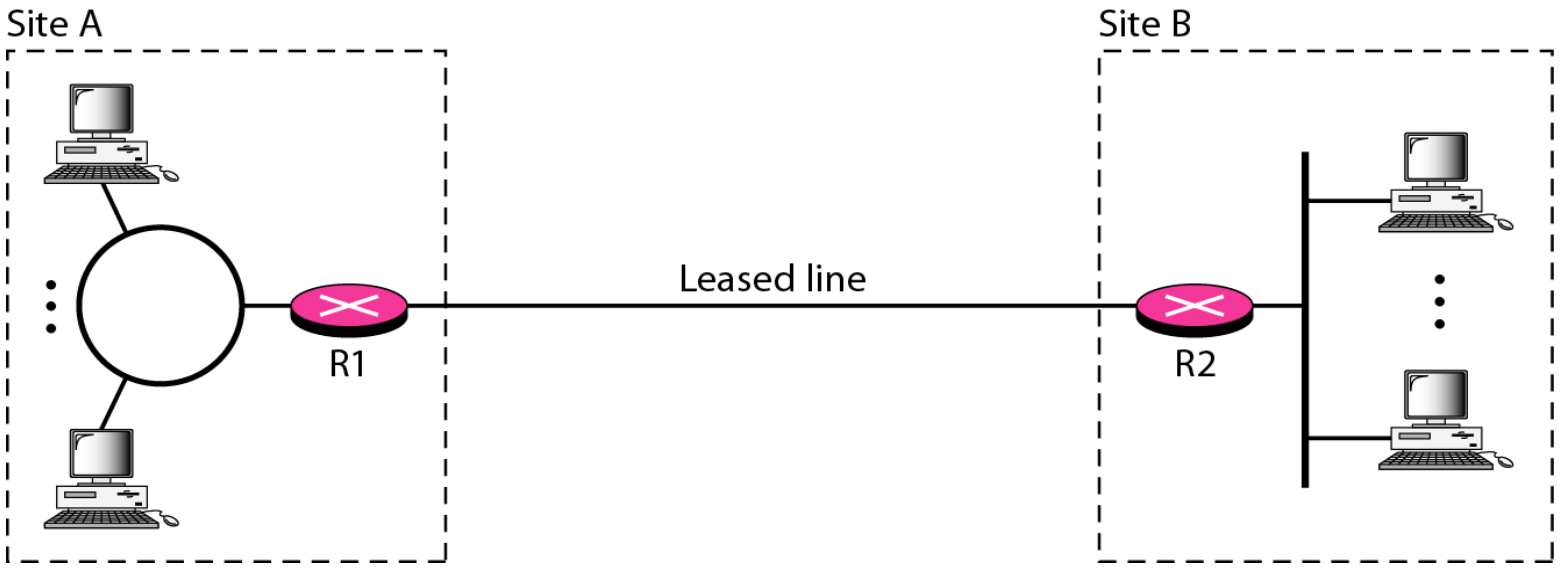
To achieve privacy, organizations can use one of three strategies:

- private networks,
- hybrid networks,
- virtual private networks.

1.Private Networks:

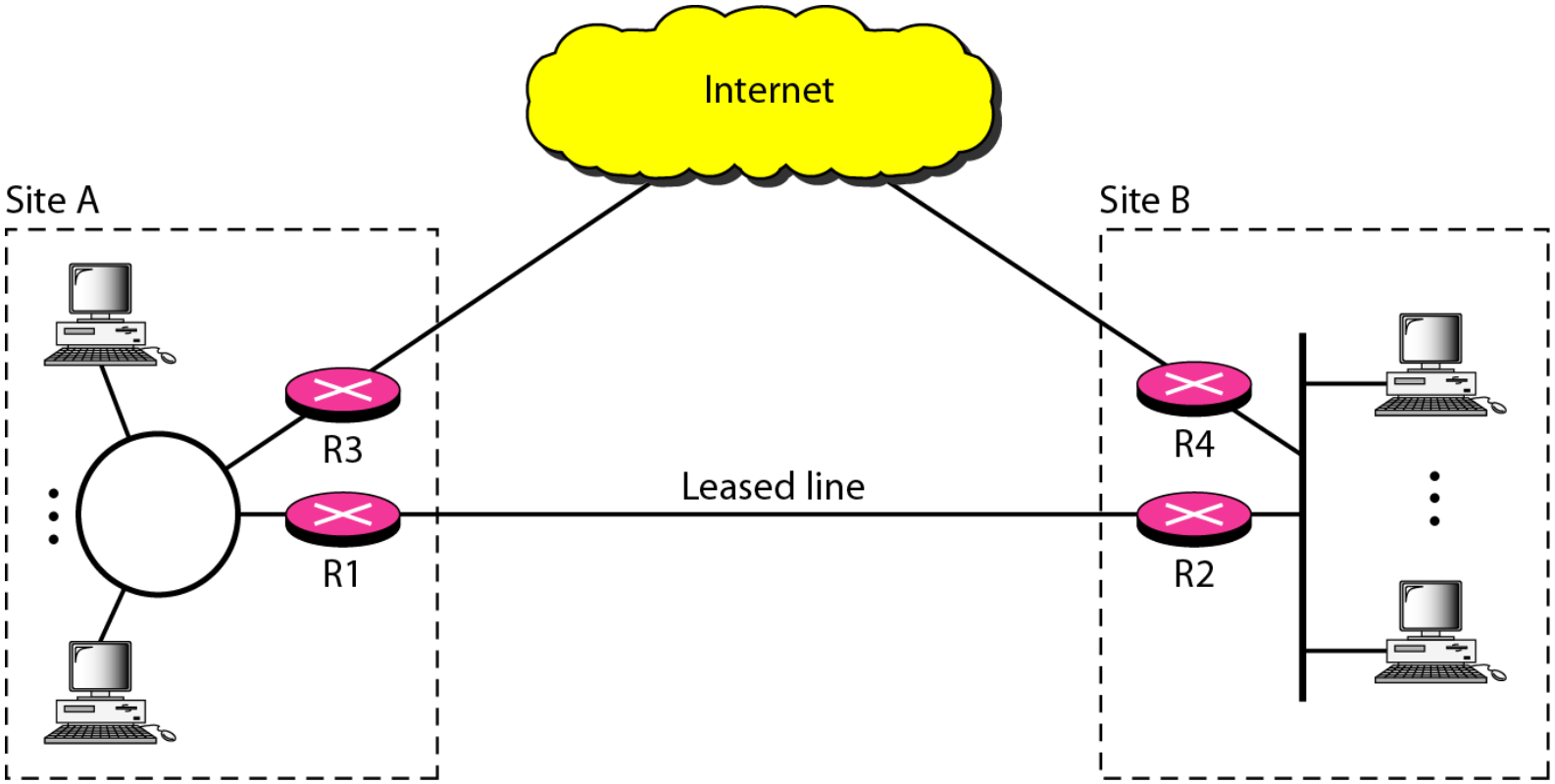
An organization that needs privacy when routing information inside the organization can use a private network as discussed previously. A small organization with one single site can use an isolated LAN. People inside the organization can send data to one another that totally remain inside the organization, secure from outsiders. A larger organization with several sites can create a private internet. The LANs at different sites can be connected to each other by using routers and leased lines. In other words, an internet can be made out of private LANs and private WANs. The following figure shows such a situation for an organization with two sites. The LANs are connected to each other by routers and one leased line.

Private network



In this situation, the organization has created a private internet that is totally isolated from the global Internet. For end-to-end communication between stations at different sites, the organization can use the Internet model. However, there is no need for the organization to apply for IP addresses with the Internet authorities. It can use private IP addresses. The organization can use any IP class and assign network and host addresses internally. Because the internet is private, duplication of addresses by another organization in the global Internet is not a problem.

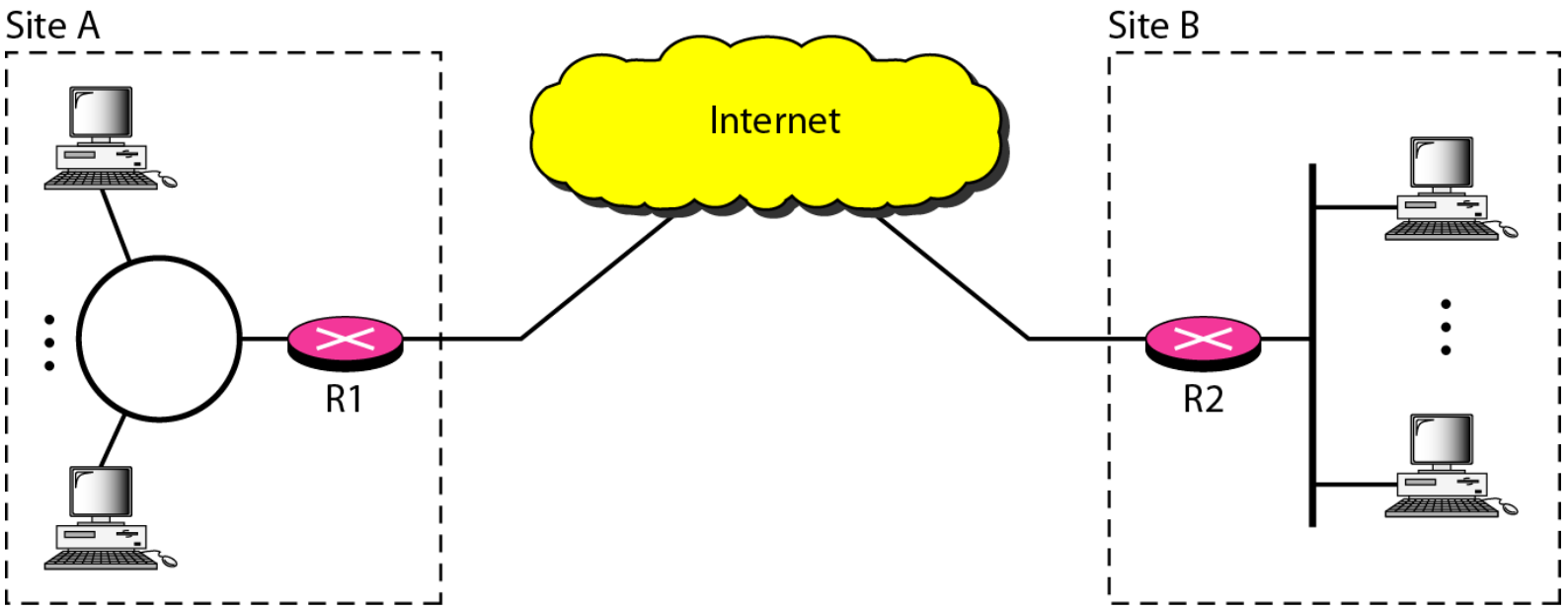
Hybrid network



Hybrid Networks Today, most organizations need to have privacy in interorganizational data exchange, but, at the same time, they need to be connected to the global Internet for data exchange with other organizations. One solution is the use of a hybrid network. A hybrid network allows an organization to have its own private internet and, at the same time, access to the global Internet. Interorganizational data are routed through the private internet; interorganizational data are routed through the global Internet.

An organization with two sites uses routers Rl and R2 to connect the two sites privately through a leased line; it uses routers R3 and R4 to connect the two sites to the rest of the world. The organization uses global IP addresses for both types of communication. However, packets destined for internal recipients are routed only through routers Rl and R2. Routers R3 and R4 route the packets destined for outsiders.

Virtual private network



3.Virtual Private Networks:

Both private and hybrid networks have a major drawback: cost. Private wide-area networks (WANs) are expensive. To connect several sites, an organization needs several leased lines, which means a high monthly fee. One solution is to use the global Internet for both private and public communications. A technology called virtual private network allows organizations to use the global Internet for both purposes.

VPN creates a network that is private but virtual. It is private because it guarantees privacy inside the organization. It is virtual because it does not use real private WANs; the network is physically public but virtually private.

VPN Technology:

VPN technology uses IPSec in the tunnel mode to provide authentication, integrity, and privacy.

Tunnelling To guarantee privacy and other security measures for an organization, VPN can use the IPSec in the tunnel mode. In this mode, each IP datagram destined for private use in the organization is encapsulated in another datagram. To use IPSec in tunnelling, the VPNs need to use two sets of addressing, as shown in following Figure.

The public network (Internet) is responsible for carrying the packet from Rl to R2. Outsiders cannot decipher the contents of the packet or the source and destination addresses. Deciphering takes place at R2, this finds the destination address of the packet and delivers it.

Addressing in a VPN

